E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Técnicas de Inteligencia Artificial para la modelización del consumo energético en un supermercado

Grado en Ingeniería Informática

Trabajo Fin de Grado

David Urdín Zaratiegui

Edurne Barrenechea Tartas

Miguel Pagola Barrio

Pamplona, 18/06/2016

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ÍNDICE

Contenido

ÍNDICE 1
FIGURAS 2
GRÁFICAS 2
ILUSTRACIONES 5
HISTOGRAMAS 5
TABLAS 5
1. MOTIVACIÓN 7
2. TÉCNICAS DE APRENDIZAJE 7
2.1 Redes Neuronales 13
2.2 Regresión Lineal 16
2.3 Arboles de decisión 20
2.4 Random Forest 22
3. TAREAS 24
3.1 Primera etapa 24
3.1.1 Algoritmo red neuronal 24
3.1.2 Algoritmo regresión lineal 26
3.1.3 Algoritmo árbol de regresión 26
3.1.4 Algoritmo random forest 27
3.1.5 Modelización de datos 27
3.2 Segunda Etapa 35
3.2.1 Supermercado Ali-Gobeo 35
3.2.2 Datos y parámetros 43
4. MARCO EXPERIMENTAL 76
5. RESULTADOS 79
6. CONCLUSIONES 84
7. BIBLIOGRAFÍA 86
8. ANEXO 87
8.1 ANEXO 1 87
8.2 ANEXO 2 94
8.3 ANEXO 3 96

FIGURAS

Figura 1: Elementos de una red neuronal y sistema global del proceso. 13
Figura 2: Estructura de una neurona artificial..... 14
Figura 3: Funciones de activación comunes..... 15
Figura 4: Estructura de un árbol de decisión con atributos discretos..... 21
Figura 5: Diagrama de caja de los valores asociados al consumo del Cuadro General de Baja Tensión del supermercado en 2015. 53
Figura 6: Estructura de un Diagrama de Caja o *Boxplot*. 54
Figura 7: Correlación entre las variables objeto de estudio del modelo representada de forma gráfica. 55
Figura 8: Ejemplo de uso de aplicación clicando en un punto. 94
Figura 9: Ejemplo de uso de la aplicación clicando y arrastrando. 95
Figura 10: Dirección, superficie y consumo del Supermercado de EROSKI del proyecto LIFE ZERO STORE..... 96

GRÁFICAS

Gráfica 1: Ejemplo de recta de regresión sobre un conjunto de datos de una única variable. .. 17
Gráfica 2: Ejemplo de distribución normal o Gaussiana de un conjunto de datos. 18
Gráfica 3: Comportamiento del parámetro R2 en train frente a test para redes neuronales de distintos tamaños. 25
Gráfica 4: Resultados obtenidos mediante el uso de una red neuronal que utiliza el 80% de los datos para entrenar y el 20% restante para realizar la predicción. 28
Gráfica 5: Resultados obtenidos mediante el uso de una red neuronal que utiliza el 70% de los datos para entrenar y el 30% restante para realizar la predicción. 29
Gráfica 6: Resultados obtenidos mediante el uso de una red neuronal que entrena y testea con un subconjunto de datos obtenido mediante reemplazamiento de boostrapping..... 30
Gráfica 7: Resultados obtenidos mediante el uso de una red neuronal que entrena y testea con un subconjunto de datos obtenido mediante la media de los conjuntos creados por boostrapping. 30
Gráfica 8: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto de datos divididos manualmente como se ha visto anteriormente. 31
Gráfica 9: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto de datos divididos mediante la técnica K-means. 32
Gráfica 10: Resultados obtenidos mediante el uso de un algoritmo de mapas auto-organizados de Kohonen. 33
Gráfica 11: Resultados obtenidos mediante el uso de un algoritmo de árbol de decisión. 34
Gráfica 12: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto determinado por la técnica de árbol de decisión anterior. 34
Gráfica 13: Consumo total por sectores..... 40
Gráfica 14: Diagrama de predicciones-realizaciones 41

Gráfica 15. Consumo eléctrico por superficie al mes a lo largo del año 2013 en diferentes zonas climáticas de establecimientos de área de superficie de ventas de aproximadamente 5000 m^2 46
Gráfica 16: Temperatura media diaria interior y temperatura media diaria exterior del año 2015..... 47
Gráfica 17: Temperatura media diaria en el exterior procedente de los analizadores de CENER y temperatura media diaria en el exterior procedente de los datos de EUSKALMET. 48
Gráfica 18: Consumo medio diario por iluminación (kWh)..... 51
Gráfica 19: Consumo diario del Cuadro General de Baja Tensión en el supermercado en 2015. 52
Gráfica 20: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos originales. 61
Gráfica 21: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos originales. 61
Gráfica 22: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos originales. 62
Gráfica 23: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos originales. 62
Gráfica 24: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos filtrados. 63
Gráfica 25: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos filtrados. 63
Gráfica 26: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos filtrados. 64
Gráfica 27: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos filtrados. 64
Gráfica 28: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados. 65
Gráfica 29: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados. 65
Gráfica 30: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados. 66
Gráfica 31: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados. 66
Gráfica 32: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados. 67
Gráfica 33: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados. 67
Gráfica 34: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados..... 68

Gráfica 35: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados..... 68
Gráfica 36: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos filtrados y añadiendo el parámetro *laboralidad.* 69
Gráfica 37: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad. 69
Gráfica 38: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad. 70
Gráfica 39: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad. 70
Gráfica 40: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando la técnica de PCA. 73
Gráfica 41: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando la técnica de PCA. 73
Gráfica 42: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando la técnica de PCA. 74
Gráfica 43: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando la técnica de PCA. 74
Gráfica 44: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales. 87
Gráfica 45: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales. 88
Gráfica 46: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales. 88
Gráfica 47: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en fines de semana y restantes y el conjunto de datos originales. 89
Gráfica 48: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales. 90
Gráfica 49: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales. 90
Gráfica 50: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales. 91
Gráfica 51: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales. 91
Gráfica 52: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos originales y añadiendo el parámetro *laboralidad*. 92
Gráfica 53: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad. 92
Gráfica 54: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad. 93

Gráfica 55: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad. 93

ILUSTRACIONES

Ilustración 1: Localización de la ciudad de Vitoria en un mapa de la península ibérica. 36
Ilustración 2: Vista aérea del supermercado de Eroski. 36
Ilustración 3. Plano del supermercado 37
Ilustración 4. Los 5 analizadores colocados en el cuadro secundario de frío. 37
Ilustración 5: Analizador de red portátiles. Ilustración 6: Medidor de temperatura portátil colocado en el CPD de CENER del mismo modo que se colocaron en el supermercado. 39
Ilustración 7: Método de retención o *Holdout.* 56
Ilustración 8: Validación cruzada de K = 4 iteraciones. 57
Ilustración 9: Validación cruzada aleatoria con K iteraciones. 58
Ilustración 10: Validación cruzada dejando uno fuera (LOOCV) 58
Ilustración 11: K-fold cross validation, con K = 4 y 4 clasificadores. 59

HISTOGRAMAS

Histograma 1: Frecuencia de los valores de la temperatura interior horaria del supermercado en el año 2015. 44
Histograma 2: Frecuencia de los valores de la temperatura exterior horaria del supermercado en el año 2015. 45
Histograma 3: Consumo de los compresores de frío positivo (kWh) 49
Histograma 4: Consumo de los compresores de frío negativo (kWh). 49
Histograma 5: Consumo de los compresores de frío positivo y negativo (kWh). 50
Histograma 6: Consumo del Cuadro General de Baja Tensión (kWh) del supermercado en 2015. 52

TABLAS

Tabla 1: Matriz de valores de componentes 71
Tabla 2: Importancia de las componentes 72
Tabla 3: Train versión 1.0 79
Tabla 4: Test versión 1.0 79
Tabla 5: Train versión 1.1 79
Tabla 6: Test versión 1.1 80
Tabla 7: Train versión 1.2 80
Tabla 8: Test versión 1.2 80
Tabla 9: Train versión 1.3 80
Tabla 10: Test versión 1.3 80
Tabla 11: Train versión 2.0 81
Tabla 12: Test versión 2.0 81
Tabla 13: Train versión 2.1 81
Tabla 14: Test versión 2.1 81
Tabla 15: Train versión 2.2 81
Tabla 16: Test versión 2.2 82

Tabla 17: Train versión 2.3 82
Tabla 18: Test versión 2.3..... 82
Tabla 19: Comparación entre los mejores de cada versión. 82
Tabla 20: Train versión 2.3 y técnica PCA. 83
Tabla 21: Test versión 2.3 y técnica PCA. 83
Tabla 22: Train versión 2.2 y técnica PCA 83
Tabla 23: Test versión 2.2 y técnica PCA. 83
Tabla 24: Train versión 2.3 y técnica de validación cruzada con K = 3..... 84
Tabla 25: Test versión 2.3 y técnica de validación cruzada con K = 3. 84

1. MOTIVACIÓN

En este proyecto se evaluarán distintos algoritmos, métodos y técnicas estadísticas para la modelización del consumo general en baja tensión de un supermercado, con el objetivo de conseguir un modelo lo suficientemente preciso (dentro de unos márgenes de error) como para predecir datos futuros de dicho consumo y establecer ahorro y poder realizar simulaciones. Se decidirá cuál se comporta mejor para el caso concreto comparando los distintos algoritmos, parámetros y resultados que intervengan en cada caso mediante tests estadísticos.

La ingente cantidad de información de la que se dispone actualmente gracias a, mayoritariamente, Internet, hace que surjan y se desarrollen nuevas tecnologías y que se tengan cada vez más en cuenta aquellas que hasta hace poco tiempo formaban parte casi en exclusividad de Universidades, centros de investigación y grandes empresas. Algunas de estas tecnologías están relacionadas con el Big Data (análisis y enfoque en la descripción y uso de grandes cantidades de datos) y con el Machine Learning (Aprendizaje Automático). Pues bien, como punto de partida, comenzaremos por explicar qué es el aprendizaje automático.

2. TÉCNICAS DE APRENDIZAJE

El aprendizaje automático o de máquinas está incluido en el ámbito de las ciencias de la computación, más en concreto en el de la Inteligencia Artificial, y tiene como objetivo la creación y desarrollo de algoritmos capaces de aprender, establecer patrones sobre datos y llegar a predecirlos. De forma más concreta, se trata de crear programas capaces de aprender a resolver problemas mediante ejemplos. Es, por lo tanto, un proceso de inducción (va desde hechos, ejemplos, a afirmaciones generales) del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático se centra más en el estudio de la complejidad computacional de los problemas. Muchos problemas son de clase NP (los problemas NP son aquellos problemas para los que no se conoce un algoritmo determinista o probabilista que los resuelva en tiempo eficiente, esto es, en tiempo polinomial o menor), por lo que gran parte de la investigación realizada está enfocada al diseño de soluciones factibles a esos problemas. El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.

El aprendizaje automático tiene una amplia gama de aplicaciones como pueden ser sugerencias y ayudas en motores de búsqueda, diagnósticos médicos, detección de fraude con tarjetas de crédito, detección de spam en el correo electrónico, reconocimiento de voz, traducción de idiomas mediante imágenes, reconocimiento de voz, detección de rostros…

No todos los sistemas de aprendizaje automático funcionan de la misma manera; en algunos se elimina toda conexión con el conocimiento que proporciona el experto acerca de los

procesos de análisis de datos, mientras que otros tratan de establecer un marco de colaboración entre el experto y la máquina. De todas formas, la intuición humana no puede ser reemplazada totalmente, ya que el diseñador del sistema ha de especificar la forma de representación de los datos, los métodos de manipulación y caracterización de los mismos y por supuesto debe comprender y evaluar cómo utilizar los resultados obtenidos. Sin embargo, las computadoras son utilizadas por todo el mundo con fines tecnológicos muy útiles.

El aprendizaje automático tiene como resultado un modelo que resuelve una tarea dada. Para la producción de dicho modelo, se tienen en cuenta los atributos y características de cada ejemplo. Así, cada instancia tendrá una configuración, un dominio, esto es, un conjunto de valores que sus atributos pueden adoptar. Los valores de dicho dominio pueden ser números, valores binarios o un conjunto de etiquetas cualquiera como el de meses, estaciones o colores.

Cómo se ha comentado anteriormente, no todos los sistemas de aprendizaje funcionan del mismo modo; la manera en que cada sistema asigna los valores a sus atributos o salidas permite precisamente clasificarlos;

- **Aprendizaje supervisado:**

  El algoritmo crea una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Básicamente se basan en la información de experiencias pasadas para aprender a definir resultados futuros. Un ejemplo de este tipo de algoritmo es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar (clasificar) una serie de vectores utilizando una entre varias categorías (clases). Este tipo de aprendizaje puede llegar a ser muy útil en problemas de investigación biológica, biología computacional y bioinformática.

- **Aprendizaje no supervisado:**

  Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema. No se tiene en cuenta información sobre ejemplos anteriores, por lo que el sistema debe ser capaz de reconocer patrones para poder reconocer las nuevas entradas.

- **Aprendizaje semi-supervisado:**

  Este tipo de algoritmos combinan los dos algoritmos anteriores para poder clasificar de manera adecuada. Se sirve tanto de instancias ya reconocidas cómo de nuevos ejemplos de los que no se tiene conocimiento anterior.

- **Aprendizaje por refuerzo:**

  El algoritmo aprende observando el mundo que le rodea. Su información de entrada es el *feedback* o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo-error.

- **Transducción:**

Similar al aprendizaje supervisado, pero no construye de forma explícita una función. Trata de predecir las categorías de los futuros ejemplos basándose en los ejemplos de entrada, sus respectivas categorías y los ejemplos nuevos al sistema.

- **Aprendizaje multi-tarea:**

  Métodos de aprendizaje que usan conocimiento previamente aprendido por el sistema de cara a enfrentarse a problemas parecidos a los ya vistos.

Distinción entre Aprendizaje supervisado y no supervisado

Son los dos más importantes (de hecho el resto provienen de alguno de los dos o de la unión de ambos) y por eso merece la pena profundizar y destacar las diferencias entre ambos. El aprendizaje supervisado se caracteriza por contar con información que especifica qué conjuntos de datos son satisfactorios para el objetivo del aprendizaje. Un ejemplo podría ser un software que reconoce si una imagen dada es o no la imagen de un rostro: para el aprendizaje del programa tendríamos que proporcionarle diferentes imágenes, especificando en el proceso si se trata o no de rostros.

En el aprendizaje no supervisado, en cambio, el programa no cuenta con datos que definan que información es satisfactoria o no. El objetivo principal de estos programas suele ser encontrar patrones que permitan separar y clasificar los datos en diferentes grupos, en función de sus atributos. Siguiendo el ejemplo anterior un software de aprendizaje no supervisado no sería capaz de decirnos si una imagen dada es un rostro o no pero sí podría, por ejemplo, clasificar las imágenes entre aquellas que contienen rostros humanos, de animales, o las que no contienen. La información obtenida por un algoritmo de aprendizaje no supervisado debe ser posteriormente interpretada por una persona para darle utilidad.

El análisis computacional y de rendimiento de los algoritmos de aprendizaje automático es una rama de la estadística conocida como teoría computacional del aprendizaje.

El aprendizaje automático las personas lo llevamos a cabo de manera inconsciente ya que es un proceso tan sencillo para nosotros que ni nos damos cuenta de cómo se realiza y todo lo que implica. Desde que nacemos hasta que morimos los seres humanos llevamos a cabo diferentes procesos, entre ellos encontramos el de aprendizaje por medio del cual adquirimos conocimientos, desarrollamos habilidades para analizar y evaluar a través de métodos y técnicas así como también por medio de la experiencia propia. Sin embargo, a las máquinas hay que indicarles cómo aprender, ya que si no se logra que una máquina sea capaz de desarrollar sus habilidades, el proceso de aprendizaje no se estará llevando a cabo, sino que solo será una secuencia repetitiva. También debemos tener en cuenta que el tener conocimiento o el hecho de realizar bien el proceso de aprendizaje automático no implica que se sepa utilizar, es preciso saber aplicarlo en las actividades cotidianas, y un buen aprendizaje también implica saber cómo y cuándo utilizar nuestros conocimientos.

Para llevar a cabo un buen aprendizaje es necesario considerar todos los factores que a este le rodean, como la sociedad, la economía, la ciudad, el ambiente, el lugar, etc. Por lo tanto, es necesario empezar a tomar diversas medidas para lograr un aprendizaje adecuado, y obtener una automatización adecuada del aprendizaje. Así, lo primero que se debe tener en cuenta es el concepto de conocimiento, que es el entendimiento de un determinado tema o materia en el cual tú puedas dar tu opinión o punto de vista, así como responder a ciertas interrogantes que puedan surgir de dicho tema o materia.

En el aprendizaje automático podemos obtener 3 tipos de conocimiento, que son:

- **Crecimiento:** Es el que se adquiere de lo que nos rodea, el cual guarda la información en la memoria como si dejara huellas.
- **Reestructuración:** Al interpretar los conocimientos el individuo razona y genera nuevo conocimiento al cual se le llama de reestructuración.
- **Ajuste:** Es el que se obtiene al generalizar varios conceptos o generando los propios.

Los tres tipos se efectúan durante un proceso de aprendizaje automático pero la importancia de cada tipo de conocimiento depende de las características de lo que se está tratando de aprender. El aprendizaje es más que una necesidad, es un factor primordial para satisfacer las necesidades de la inteligencia artificial.

El aprendizaje automático está englobado por la minería de datos (es la etapa de análisis de "Knowledge Discovery in Databases" o KDD, el proceso completo de extracción de conocimiento a partir de bases de datos). Se trata de un proceso de extracción de conocimiento a partir de grandes cantidades de datos. Utiliza métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos para explorar y analizar grandes cantidades de datos y descubrir patrones significativos, es decir, consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior.

La tarea de minería de datos real es el análisis automático o semi-automático de grandes cantidades de datos para extraer patrones interesantes hasta ahora desconocidos, como los grupos de registros de datos (análisis clúster), registros poco usuales (la detección de anomalías) y dependencias (minería por reglas de asociación). Esto generalmente implica el uso de técnicas de bases de datos como los índices espaciales. Estos patrones pueden entonces ser vistos como una especie de resumen de los datos de entrada, y pueden ser utilizados en el análisis adicional o, por ejemplo, en la máquina de aprendizaje y análisis predictivo. Por ejemplo, el paso de minería de datos podría identificar varios grupos en los datos, que luego pueden ser utilizados para obtener resultados más precisos de predicción por un sistema de soporte de decisiones. Ni la recolección de datos, preparación de datos, ni la interpretación de los resultados y la información son parte de la etapa de minería de datos, pero que pertenecen a todo el proceso KDD como pasos adicionales.

Los términos relacionados con la obtención de datos, la pesca de datos y espionaje de los datos se refieren a la utilización de métodos de minería de datos a las partes de la muestra de un conjunto de datos de población más grandes establecidas que son (o pueden ser) demasiado pequeñas para las inferencias estadísticas fiables que se hizo acerca de la validez de cualquier patrón descubierto. Estos métodos pueden, sin embargo, ser utilizados en la creación de nuevas hipótesis que se prueban contra poblaciones de datos más grandes.

Un proceso típico de KDD consta de los siguientes pasos generales:

- **Selección del conjunto de datos**, tanto en lo que se refiere a las variables objetivo (aquellas que se quiere predecir, calcular o inferir), como a las variables independientes (las que sirven para hacer el cálculo o proceso), como posiblemente al muestreo de los registros disponibles.
- **Análisis de las propiedades de los datos**, en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).

- **Transformación del conjunto de datos de entrada**, se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema, a este paso también se le conoce como pre procesamiento de los datos.
- **Seleccionar y aplicar la técnica de minería de datos**, se construye el modelo predictivo, de clasificación o segmentación.
- **Extracción de conocimiento**, mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un pre procesado diferente de los datos.
- **Interpretación y evaluación de datos**, una vez obtenido el modelo, se debe proceder a su validación comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

Si el modelo final no superara esta evaluación el proceso se podría repetir desde el principio o, si el experto lo considera oportuno, a partir de cualquiera de los pasos anteriores. Esta retroalimentación se podrá repetir cuantas veces se considere necesario hasta obtener un modelo válido. Una vez validado el modelo, si resulta ser aceptable (proporciona salidas adecuadas y/o con márgenes de error admisibles) éste ya está listo para su explotación.

Tradicionalmente, las técnicas de minería de datos se aplicaban sobre información contenida en almacenes de datos. De hecho, muchas grandes empresas e instituciones han creado y alimentan bases de datos especialmente diseñadas para proyectos de minería de datos en las que centralizan información potencialmente útil de todas sus áreas de negocio. No obstante, actualmente está cobrando una importancia cada vez mayor la minería de datos desestructurados como información contenida en ficheros de texto, en Internet… ya que la tecnología, herramientas, software, hardware… han progresado de manera que ya no se ven obstaculizados por la capacidad de recopilar datos, sino por la capacidad de gestionar, analizar, sintetizar, visualizar y, en resumen, descubrir el conocimiento de los datos recopilados de manera oportuna y eficaz.

Como ya se ha comentado, las técnicas de la minería de datos provienen de la inteligencia artificial, aprendizaje automático, estadística… dichas técnicas, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener unos resultados.

A continuación se proporciona una breve descripción de algunas de las más representativas, y posteriormente se desarrollará y profundizará en aquellas que se vayan a utilizar en el proyecto.

- **Árbol de decisión:** Un árbol de decisión es una técnica que en función de un conjunto de atributos permite determinar a qué clase pertenece o qué valor adquiere el caso objeto de estudio. En cada nodo de decisión se especifica una prueba o test a realizar y los posibles resultados de la prueba en cuestión son los descendientes del nodo. Puede haber más de un árbol de decisión correcto para un mismo conjunto de datos

dependiendo del orden en el que se van tomando los atributos. Se suele elegir cada vez el atributo que mejor clasifica, hecho que nos indica una medida denominada impureza del nodo.

- **Reglas de asociación:** Los algoritmo de reglas de asociación tratan de descubrir relaciones o patrones interesantes entre las distintas variables. Clasifican nuevos ejemplos basándose en el conocimiento aprendido. Una regla está compuesta por dos elementos; antecedente, que contiene un predicado que se evaluará como verdadero o falso para cada ejemplo, y el consecuente, que contiene una etiqueta de clase o valor numérico que adquiere el valor a predecir. A diferencia de los árboles, con las reglas no hay un orden determinado para realizar la búsqueda.

- **Algoritmos genéticos:** Los algoritmos genéticos son algoritmos adaptativos de optimización, búsqueda y/o aprendizaje que están inspirados en la selección natural y evolución genética. Utilizan métodos tales como la mutación y el cruzamiento para generar nuevas clases que puedan ofrecer una buena solución a un problema dado.

- **Redes neuronales artificiales:** Las redes de neuronas artificiales (RNA) son un modelo de aprendizaje automático inspirado en las neuronas de los sistemas nerviosos de los animales. Se trata de un sistema de enlaces de neuronas que colaboran entre sí para producir un estímulo de salida. Las conexiones tienen pesos numéricos que se adaptan según la experiencia. De esta manera, las redes neurales se adaptan a un impulso y son capaces de aprender. El modelo más básico de RNA se denomina perceptrón. A partir de este se pueden desarrollar sistemas más complejos denominados multicapa.

- **Algoritmo de agrupamiento:** El agrupamiento es un método de aprendizaje no supervisado y es una técnica muy popular de análisis estadístico de datos. Es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes. Ejemplos:
  - Algoritmo K-means.
  - Algoritmo K-medoids

- **Redes bayesianas:** Una red bayesiana, red de creencia o modelo acíclico dirigido es un modelo probabilístico que representa una serie de variables de azar y sus independencias condicionales a través de un grafo acíclico dirigido. Una red bayesiana puede representar, por ejemplo, las relaciones probabilísticas entre enfermedades y síntomas. Dados ciertos síntomas, la red puede usarse para calcular las probabilidades de que ciertas enfermedades estén presentes en un organismo. Hay algoritmos eficientes que infieren y aprenden usando este tipo de representación.

- **Regresión lineal simple y múltiple:** Es de las más utilizadas para formar relaciones entre datos ya que es rápida y eficaz. La simple es insuficiente en espacios multidimensionales donde puedan relacionarse más de 2 variables. Se puede utilizar para este caso la regresión lineal múltiple. Actualmente existen técnicas más modernas y precisas que hacen que la regresión no sea la mejor opción, pero siempre conviene tenerla en cuenta ya que es una técnica muy fiable, simple y utilizada a lo largo de la historia.

- **Modelos estadísticos**.- Es una expresión simbólica en forma de igualdad o ecuación que se emplea en todos los diseños experimentales y en la regresión para indicar los diferentes factores que modifican la variable de respuesta.

A continuación se explicará más en profundidad las técnicas y/o algoritmos utilizados en el proyecto.

2.1 Redes Neuronales

En primer lugar es necesario explicar brevemente qué es una red neuronal y para qué se utiliza, debido a que es el algoritmo principal del proyecto.

Las **redes de neuronas artificiales** (denominadas habitualmente como **RNA** o en inglés como "ANN") son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como **redes de neuronas** o **redes neuronales**. La base de los ANS (Artificial Neural Systems) imita la estructura hardware del sistema nervioso, con la intención de construir sistemas de procesamiento de información paralelos, distribuidos y adaptativos, que puedan presentar un cierto comportamiento inteligente. En un sistema neuronal biológico, los elementos básicos son las neuronas, que se agrupan en conjuntos compuestos por millones de ellas organizadas en capas, constituyendo un sistema con funcionalidad propia. Un conjunto de esos subsistemas da lugar a un sistema global, el sistema nervioso. En la realización de un sistema neuronal artificial, puede establecerse una estructura similar. El elemento esencial de partida es la neurona artificial, que se organiza en capas, varias capas constituyen una red neuronal y, por último, una red neuronal junto con las interfaces de entrada y salida, más los módulos algorítmicos adicionales necesarios, conforman el sistema global de proceso (Figura 1).

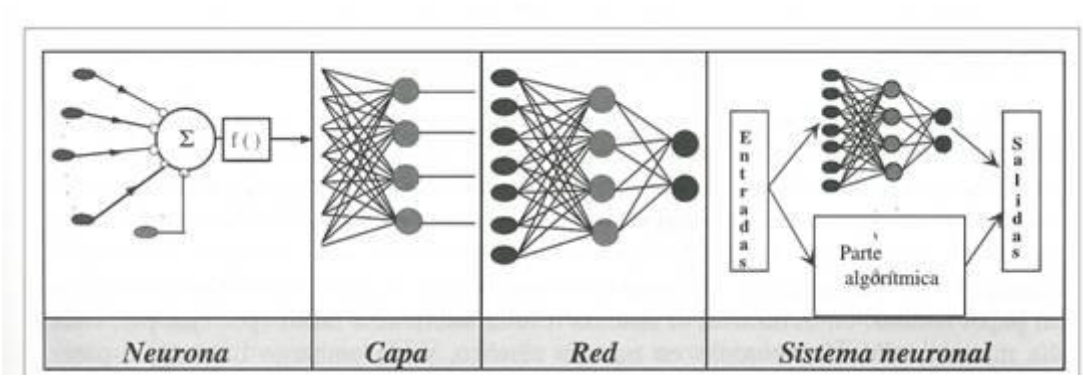


Figura 1: Elementos de una red neuronal y sistema global del proceso.

Modelo general de neurona artificial

En este apartado se describe la estructura genérica de neurona artificial. Se denomina procesador elemental o neurona a un dispositivo simple de cálculo que a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida. Los elementos que constituyen la neurona son los siguientes (Figura 2):

1. Conjunto de entradas, $x_j\ (t)$
2. Pesos sinápticos de la neurona *i*, w_{ij} que representan la intensidad de interacción entre cada neurona presináptica *j* y la neurona postsináptica *i*.
3. Reglas de propagación $\sigma(w_{ij}, x_j\ (t))$ que proporciona el valor del potencial postsináptico $h_i\ (t) = \sigma(w_{ij}, x_j\ (t))$ de la neurona *i* en función de sus pesos y entradas.
4. Función de activación $f_i\ (a_i\ (t-1), h_i\ (t))$ que indica el estado de activación actual $a_i\ (t) = f_i\ (a_i\ (t-1), h_i\ (t))$ de la neurona *i*, en función de su estado actual $a_i\ (t-1)$ y de su potencial postsináptico actual. Puede no existir siendo en este caso la salida la misma función de propagación, $a_i\ (t) = \sigma(w_{ij}, x_j\ (t))$.
5. Función de salida $F_i\ (a_i\ (t))$ que proporciona la salida actual $y_i\ (t) = F_i\ (a_i\ (t))$ de la neurona *i* en función de su estado de activación. De este modo la activación de la neurona *i* puede expresarse como $y_i\ (t) = F_i\ (f_i\ [a_i\ (t - 1), \sigma(w_{ij}, x_j\ (t))])$

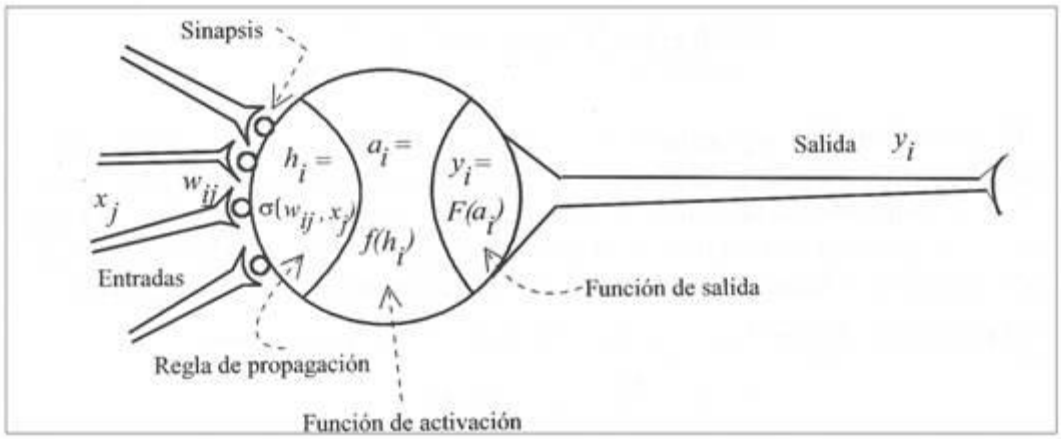


Figura 2: Estructura de una neurona artificial.

A continuación se profundiza en cada uno de los elementos que componen una neurona.

Entradas y salidas

Las variables de entrada y salida pueden ser binarias (digitales) o continuas (analógicas), dependiendo del modelo y aplicación. Dependiendo del tipo de salida, las neuronas suelen recibir nombres específicos. Así, las neuronas estándar cuya salida sólo puede tomar los valores 0 ó 1 se suelen denominar genéricamente neuronas tipo McCulloch-Pitts, mientras que aquéllas que únicamente pueden tener por salidas -1 ó +1 se suelen denominar neuronas tipo Ising. Si puede adoptar diversos valores discretos en la salida (por ejemplo, -2, -1, 0, 1, 2), se dice que se trata de una neurona de tipo Potts. Por otro lado, en el caso de las analógicas puede ocurrir que una neurona de salida continua, que puede proporcionar valores cualquiera, se limite a un intervalo definido, por ejemplo, [0, +1] ó [-1, +1].

Regla de propagación

La regla de propagación permite obtener, a partir de las entradas y los pesos, el valor del potencial postsináptico h_i de la neurona

$$h_i\ (t) = \sigma_i\ (w_{ij}, x_j\ (t))$$

La función más habitual es de tipo lineal y se basa en la suma ponderada de las entradas con los pesos sinápticos;

$$h_i\ (t) = \sum w_{ij}\ x_j$$

El peso sináptico w_{ij} define, en este caso, la intensidad de interacción entre la neurona presináptica j y la postsináptica i. Dada una entrada positiva, si el peso es positivo tenderá a excitar a la neurona, si el peso es negativo tenderá a inhibirla.
Otras reglas de tipo no lineal, también son utilizadas en la literatura, al igual que otras basadas en las distancia entre vectores, por ejemplo la denominada distancia euclidiana (siendo dos puntos de un espacio geométrico *x* e *y,* la distancia euclidiana entre ambos puntos será: $d(x, y) = sqrt((x – y)^2))$

Función de activación o función de transferencia.
La función de activación o de transferencia proporciona el estado de activación actual $a_i\ (t)$ a partir del potencial postsináptico $h_i(t)$ y del propio estado de activación anterior $a_i(t -1)$.

$$a_i\ (t) = f_i\ (a_i\ (t\text{-}1),\ h_i\ (t))$$

También, existen muchos modelos de ANS donde el estado actual de la neurona no depende de su estado anterior, sino únicamente del actual.
La función de activación f( ) se suele considerar determinista, y en la mayor parte de los modelos es monótona creciente y continua. Las formas de las funciones de activación más empleadas en los ANS se muestran en la Figura 3, donde se ha denotado con "x" al potencial postsináptico y con "y" al estado de activación. La más simple de todas es la función identidad. Otro caso también muy simple es la función escalón, empleada en el perceptrón simple.

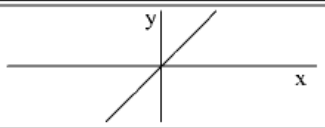
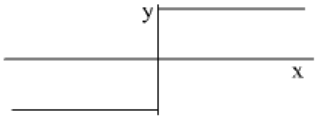
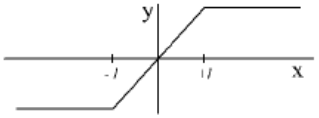
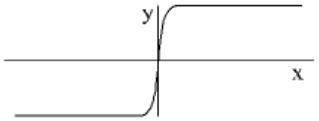
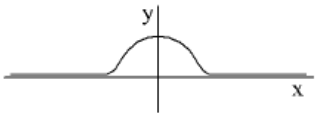
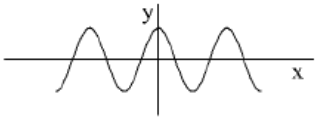
	Función	Rango	Gráfica
Identidad	$y=x$	$[-\infty, +\infty]$	
Escalón	$y = \text{signo}(x)$ $y = H(x)$	$[-1, +1]$ $[0, +1]$	
Lineal a tramos	$y = \begin{cases} -1, si\ x < -1 \\ x, si\ +1 \le x \le -1 \\ +1, si\ x > +1 \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A\,\text{sen}(wx + \varphi)$	$[-1, +1]$	

Figura 3: Funciones de activación comunes.

Función de salida
Esta función proporciona la salida global de la neurona $y_i(t)$ en función de su estado de activación actual $a_i\ (t)$. Muy frecuentemente, la función de salida es simplemente la identidad F(x) = x, de modo que el estado de activación de la neurona se considera como la propia salida

$$Y_i(t) = F_i\ (a_i\ (t)) = a_i\ (t)$$

Redes neuronales supervisadas

Uno de los modelos de redes neuronales más populares son las redes neuronales supervisadas. En este tipo de redes, con aprendizaje supervisado, se presenta a la red un conjunto de patrones, junto con la salida deseada u objetivo, e iterativamente, ésta ajusta sus pesos hasta que su salida tiende a ser la deseada, utilizando para ello información detallada del error que se comete en cada paso. De este modo, la red es capaz de estimar relaciones entrada-salida sin necesidad de proponer una cierta forma funcional de partida.

Así, dentro de este grupo de redes, denominadas redes unidireccionales organizadas en capas con aprendizaje supervisado, se puede hablar del asociador lineal, perceptrón simple, adalina y perceptrón multicapa. En este último tipo, habitualmente se aplica un algoritmo de aprendizaje denominado back-propagation (retropropagación) o BP que es el utilizado en este trabajo. Cada uno de ellos es una evolución del anterior.

Al añadir capas intermedias (ocultas) a un perceptrón simple, se obtiene un perceptrón multicapa o MLP (Multi-Layer Perceptron). Esta arquitectura suele entrenarse, como se ha dicho, mediante el algoritmo denominado de retropropagación de errores o BP. Así, el conjunto arquitectura MLP + aprendizaje BP suele denominarse red de retropropagación, o simplemente BP.

Se denota x_i a las entradas de la red, y_i a las salidas de la capa oculta, z_k a las de la capa final y t_k a las salidas objetivo, además, sean w_{ij} los pesos de la capa oculta y θ_j sus umbrales, w'_{ij} los pesos de la capa de salida y θ'_j sus umbrales. La operación de un MLP con una capa oculta y neuronas de salida lineal se expresa matemáticamente de la siguiente manera:

$$z_k = \sum_j w'_{kj}\, y_j - \theta'_k = \sum_j w'_{kj}\, f\left(\sum_i w_{ji} x_i - \theta_j\right) - \theta'_k$$

Siendo f() de tipo sigmoide;

$$f(x) = \frac{1}{1+e^{-x}}$$

2.2 Regresión Lineal

La **regresión lineal** o **ajuste lineal** es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente *Y*, las variables independientes X_i y un término aleatorio ε. Este modelo puede ser expresado como:

$$yi = B0 + B1xi1 + B2xi2 + \cdots + BP\, xiP + ei,$$

dónde:

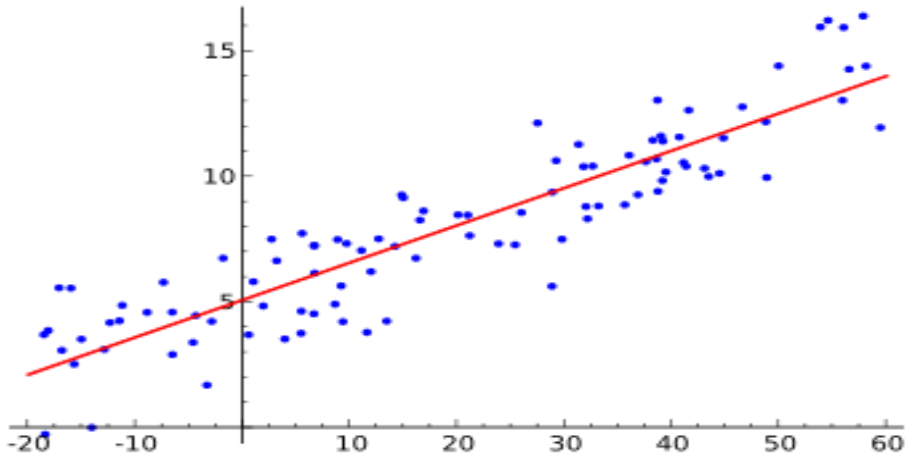
Y_i : variable dependiente, explicada.

$X_1 \ldots X_p$: variables explicativas, independientes.

$B_1 \ldots B_p$: parámetros, miden la influencia que las variables explicativas tienen sobre la independiente. El término B_0 es la intersección o término "constante", las B_i (i>0) son los parámetros respectivos a cada variable independiente, y *P* es el número de parámetros independientes a tener en cuenta en la regresión. La regresión lineal puede ser contrastada con la regresión no lineal (cuando la relación entre X e Y tiene

algún grado de curvatura; algunos ejemplos son la exponencial $Y = AX^b$ y la logarítmica $log(Y) = log(A) + b.log(X))$

El término *lineal* se emplea para distinguirlo del resto de técnicas de regresión, que emplean modelos basados en cualquier clase de función matemática. Los modelos lineales son una explicación simplificada de la realidad, mucho más ágiles y con un soporte teórico mucho más extenso por parte de la matemática y la estadística. Muchas veces lo más sencillo es lo mejor. La regresión lineal solo funciona por tanto correctamente cuando la relación entre variable dependiente e independientes es lineal, generan un hiperplano. Cuando solo existe una variable, la relación es una recta. Si tiene más, será un plano.



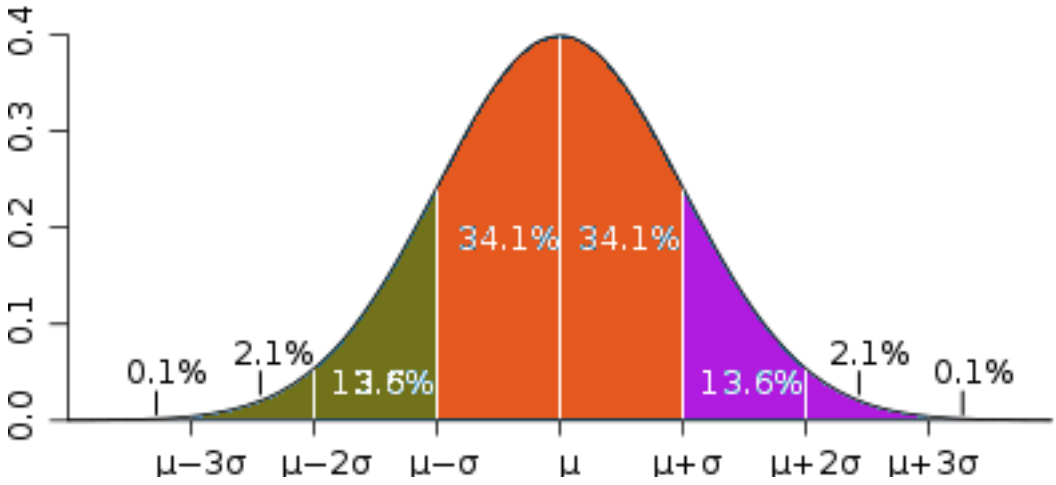
Gráfica 1: Ejemplo de recta de regresión sobre un conjunto de datos de una única variable.

El término *regresión* se utilizó por primera vez en el estudio de variables antropométricas: al comparar la estatura de padres e hijos, donde resultó que los hijos cuyos padres tenían una estatura muy superior al valor medio, tendían a igualarse a éste, mientras que aquellos cuyos padres eran muy bajos tendían a reducir su diferencia respecto a la estatura media; es decir, "regresaban" al promedio. La constatación empírica de esta propiedad se vio reforzada más tarde con la justificación teórica de ese fenómeno.

Hipótesis del modelo de regresión lineal clásico

1. **Esperanza matemática nula**: $\mathbb{E}(\varepsilon_i) = 0$. Para cada valor de X la perturbación tomará distintos valores de forma aleatoria, pero no tomará sistemáticamente valores positivos o negativos, sino que se supone tomará algunos valores mayores que cero y otros menores que cero, de tal forma que su valor esperado sea cero.
2. **Homocedasticidad**: Varianza constante. $\mathrm{Var}(\varepsilon_t) = \mathbb{E}(\varepsilon_t - \mathbb{E}\varepsilon_t)^2 = \mathbb{E}\varepsilon_t^2 = \sigma^2$ para todo *t*. Todos los términos de la perturbación tienen la misma varianza que es desconocida. La dispersión de cada ε_t en torno a su valor esperado es siempre la misma.
3. **Incorrelación o independencia**: No existe correlación. $\mathrm{Cov}(\varepsilon_t, \varepsilon_s) = (\varepsilon_t - \mathbb{E}\varepsilon_t)(\varepsilon_s - \mathbb{E}\varepsilon_s) = \mathbb{E}\varepsilon_t\varepsilon_s = 0$ para todo *t,s* con *t* distinto de *s*. Las covarianzas entre las distintas pertubaciones son nulas, lo que quiere decir que no están correlacionadas. Esto implica que el valor de la perturbación para cualquier observación muestral no viene influenciado por los valores de las perturbaciones correspondientes a otras observaciones muestrales.

4. **Regresores no estocásticos**: Regresores deterministas, que no dependen de valores aleatorios.
5. **Independencia lineal**. No existen relaciones lineales exactas entre los regresores.
6. $T > k + 1$. Suponemos que no existen errores de especificación en el modelo, ni errores de medida en las variables explicativas.
7. **Normalidad** de las perturbaciones: $\varepsilon \sim N(0, \sigma^2)$



Gráfica 2: Ejemplo de distribución normal o Gaussiana de un conjunto de datos.

Tipos de modelos de regresión lineal

Existen diferentes tipos de regresión lineal que se clasifican de acuerdo a sus parámetros:

Regresión lineal simple

Cuando únicamente interviene una variable independiente en el cálculo de la variable a predecir estamos ante regresión lineal simple. Matemáticamente se puede escribir como:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

donde ε_i es el error asociado a la medición del valor X_i.

Las variables B0 y B1 son desconocidas y se debe obtener un valor para cada una de ellas mediante instancias (Xi, Yi), de manera que lleguemos a obtener *y = β0 + β1x,* dónde Y es la predicción. Como hemos dicho, ε_i = Yi – Y'i.

Definimos la suma de residuales al cuadrado (RSS) cómo RSS = $e_1^2 + e_2^2 + \ldots + e_n^2$ o equivalentemente

$$\mathrm{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \ldots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

Los parámetros B0 y B1 buscados minimizan el RSS, por lo que realizando diversos cálculos obtenemos;

Dado el modelo de regresión simple anterior, si se calcula la esperanza (valor esperado) del valor *Y*, se obtiene:

$$E(y_i) = \hat{y}_i = E(\beta_0) + E(\beta_1 x_i) + E(\varepsilon_i)$$

Derivando respecto a $\hat{\beta}_0$ y $\hat{\beta}_1$ e igualando a cero, se obtiene:

$$\frac{\partial \sum (y_i - \hat{y}_i)^2}{\partial \hat{\beta}_0} = 0$$

$$\frac{\partial \sum (y_i - \hat{y}_i)^2}{\partial \hat{\beta}_1} = 0$$

Obteniendo dos ecuaciones denominadas ecuaciones normales que generan la siguiente solución para ambos parámetros:

$$\hat{\beta}_1 = \frac{\sum x \sum y - n \sum xy}{(\sum x)^2 - n \sum x^2} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

$$\hat{\beta}_0 = \frac{\sum y - \hat{\beta}_1 \sum x}{n} = \bar{y} - \hat{\beta}_1 \bar{x}$$

La interpretación del parámetro medio β_1 es que un incremento en Xi de una unidad, Yi incrementará en β_1

Regresión lineal múltiple

La regresión lineal permite trabajar con una variable a nivel de intervalo o razón. De la misma manera, es posible analizar la relación entre dos o más variables a través de ecuaciones, lo que se denomina **regresión múltiple** o **regresión lineal múltiple**.

Constantemente en la práctica de la investigación estadística, se encuentran variables que de alguna manera están relacionadas entre sí, por lo que es posible que una de las variables puedan relacionarse matemáticamente en función de otra u otras variables.

Maneja varias variables independientes. Cuenta con varios parámetros. Se expresan de la forma:

$$Y_i = \beta_0 + \sum \beta_i X_{ip} + \varepsilon_i$$

donde ε_i es el error asociado a la medición i del valor X_{ip} y siguen los supuestos de modo que $\varepsilon_i \sim N(0, \sigma^2)$ (media cero, varianza constante e igual a un σ y $\varepsilon_i \perp \varepsilon_j$ con $i \neq j$).

2.3 Arboles de decisión

Un árbol de decisión es un modelo de predicción que funciona reduciendo el espacio de predictores a un número más simple de regiones. Dados una serie de datos, se fabrican diagramas de construcciones lógicas que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva. Esta serie de reglas que dividen los datos de forma lógica se pueden representar en forma de árbol, de ahí el nombre. Los modelos basados en árboles son simples y fácilmente interpretables, pero tienen dos puntos débiles destacables; pequeños cambios en los datos pueden cambiar drásticamente la estructura del árbol y además no son tan competitivos, en términos de precisión, cómo otros modelos generados con técnicas de redes neuronales, por ejemplo. Existen varias técnicas como el bagging, random forest y boosting, las cuales veremos más adelante, en las que se utilizan múltiples árboles consiguiendo un gran incremento en la precisión pero perdiendo interpretabilidad.

Un árbol de decisión está compuesto por varios elementos:

- **Entradas:** Al igual que en cualquier otro modelo de predicción, un árbol recibe una serie de datos de entrada a partir de los cuales obtendrá una serie de reglas o patrones que se utilizarán para llevar a cabo la predicción. Los valores que pueden tomar las entradas pueden ser valores discretos o continuos.
- **Salida:** La respuesta obtenida. Puede ser un valor discreto o un valor continuo.
- **Nodo de decisión:** Es un punto en el que se debe tomar una decisión. Dependiendo de la regla obtenida, se seguirá un camino u otro. Por ejemplo: si x > 2, entonces un camino, si x <= 2 entonces el otro. Este tipo de nodos es cuadrado. El nodo superior en un árbol se denomina nodo raíz.
- **Nodo de probabilidad:** Indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema. Este tipo de nodos es redondo.
- **Nodo hoja:** Representa un estado final, esto es, un valor de la variable de destino. Son aquellos nodos que aparecen al final de cada rama, cuando el árbol deja de dividirse.
- **Arco:** se utiliza para conectar dos nodos entre sí. Los arcos procedentes de un nodo etiquetado con una característica determinada están etiquetados con cada uno de los posibles valores de dicha característica.
- **Rama:** Conjunto de nodos y arcos que forman un posible camino.

Un árbol de decisión lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. En definitiva, una representación en forma de árbol contiene ramas que se bifurcan en función de los valores tomados por las variables y que terminan en una acción concreta.

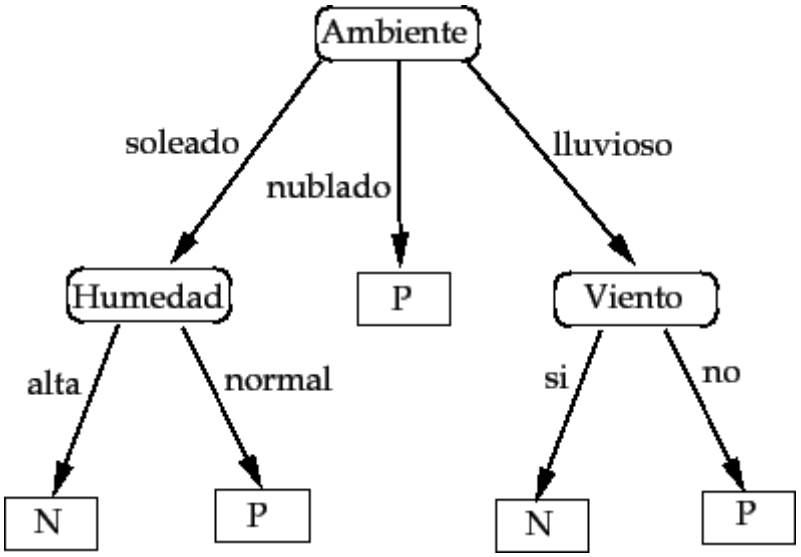


Figura 4: Estructura de un árbol de decisión con atributos discretos.

Un árbol puede ser "aprendido" mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor de atributo. Este proceso se repite en cada subconjunto derivado de una manera recursiva llamada particionamiento recursivo. La recursividad termina cuando el subconjunto en un nodo tiene todo el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones. Este proceso de *inducción top-down de los árboles de decisión* (ITDAD) es un ejemplo de un algoritmo voraz, y es, con mucho, la estrategia más común para aprender árboles de decisión a partir de datos.

En minería de datos, los árboles de decisión se pueden describir también como la combinación de técnicas matemáticas y computacionales para ayudar a la descripción, la categorización y la generalización de un conjunto dado de datos.

Los datos provienen en registros de la forma:

$$(\mathbf{x}, Y) = (x_1, x_2, x_3, ..., x_k, Y)$$

La variable dependiente, Y, es la variable objetivo que estamos tratando de entender, clasificar o generalizar. El vector **x** se compone de las variables de entrada, x_1, x_2, x_3 etc., que se utilizan para esa tarea.

Los árboles de decisión utilizados en la minería de datos son de dos tipos principales:

- **Árboles de clasificación** es cuando el resultado predicho es la clase a la que pertenecen los datos.
- **Árboles de regresión** es cuando el resultado predicho se puede considerar un número real (por ejemplo, el precio de una casa, o la longitud de la estancia de un paciente en un hospital).

El término **Árboles de Clasificación y Regresión** (ACR) es un término genérico utilizado para referirse a ambos de los procedimientos anteriores, introducido por primera vez por Breiman. Los árboles utilizados para la regresión y los árboles utilizados para la clasificación tienen algunas similitudes - pero también algunos diferencias, tales como el procedimiento utilizado para determinar donde dividir.

Algunas técnicas que ya hemos nombrado, a menudo llamados métodos conjuntos híbridos, construyen más de un árbol de decisión:

- **Bagging:** un método de conjunto, construye múltiples árboles de decisión haciendo repetidamente remuestreo de los datos de entrenamiento con sustitución, y votando los árboles para hallar una predicción de consenso.
- **Random Forest:** utiliza una serie de árboles de decisión con el fin de mejorar la tasa de clasificación de manera similar al bagging. En cada bifurcación de un árbol, las variables que se tienen en cuenta son *m,* siendo *m* un número aleatorio. (posteriormente se analizará esta técnica más en detalle)
- **Boosting:** se pueden utilizar para problemas de regresión y de clasificación. Es similar al bagging, pero los árboles crecen de manera secuencial. Cada árbol crece utilizando información de árboles anteriores.

Hay muchos algoritmos específicos de árbol de decisiones. Entre los más destacados están:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (Sucesor de ID3)
- ACR (Árboles de Clasificación y Regresión)
- CHAID (Detector automático de Chi-cuadrado de interacción). Realiza divisiones de múltiples niveles al calcular los árboles de clasificación.
- MARS: Extiende los árboles de decisión para manejar mejor datos numéricos.
- Árboles de Inferencia Condicional. Enfoque que utiliza pruebas no paramétricas como criterios de división, corregidos para múltiples pruebas para evitar el sobreajuste. Este enfoque se traduce en la selección de un predictor imparcial y no requiere poda.

ID3 y ACR se inventaron de forma independiente en la misma época (entre 1970 y 1980) pero ambos siguen un enfoque similar para el aprendizaje basado en árboles de decisión a partir de tuplas de entrenamiento.

2.4 Random Forest

Cómo se ha mencionado anteriormente, la técnica Random Forest (bosques aleatorios) es un tipo de *ensemble*. Un *ensemble* es un clasificador que está formado por muchos clasificadores llamados clasificadores base. Se aprende cada clasificador base. Para clasificar un ejemplo se utiliza cada clasificador base y se agregan las salidas para determinar la clase del ejemplo. El principio principal de los *ensembles* es unir un grupo de clasificadores “débiles” para formar uno “fuerte”. En los random forests los clasificadores base son árboles de decisión.

En muchos problemas el rendimiento del algoritmo random forest es muy similar a la del boosting, y es más simple de entrenar y ajustar. Como una consecuencia el random forests es popular y es ampliamente utilizado.

La idea esencial del bagging es promediar muchos modelos ruidosos pero aproximadamente imparciales, y por tanto reducir la variación. Los árboles son los candidatos ideales para el bagging, dado que ellos pueden registrar estructuras de interacción compleja en los datos, y si

crecen suficientemente, tienen relativamente baja parcialidad. Debido a que los árboles son notoriamente ruidosos, se benefician notoriamente al promediar.

El aprendizaje de un random forest es el siguiente:

1. Se selecciona el número de árboles que lo compondrán.
2. Para cada árbol, se realizan varios pasos;
3. Se seleccionan *N* ejemplos con reemplazamiento del conjunto de entrenamiento, siendo *N* el número de ejemplos.
4. En cada nodo se eligen aleatoriamente *m* atributos. *m* debe ser mucho menor que *M* (número total de atributos).
5. Los ejemplos que no se hayan seleccionado se utilizan para estimar el error.
6. Calcular la mejor partición a partir de las *m* variables del conjunto de entrenamiento.

Para la predicción de un nuevo caso, éste se clasifica con cada uno de los árboles que componen el random forest y se anota la etiqueta o valor del nodo hoja donde termina. La etiqueta o valor que obtenga la mayor cantidad de incidencias es reportada como la predicción. El error de un random forest depende de la correlación entre cualquier par de árboles y de la calidad individual de cada árbol que lo compone.

Tienen un buen rendimiento, son relativamente robustos frente al ruido y *outliers*, son rápidos y simples, pero no tienen la interpretabilidad de los árboles de decisión.

3. TAREAS

Como consecuencia de una colaboración con CENER comenzamos este proyecto. Para explicar el desarrollo y contenido del trabajo desempeñado, hemos decidido dividirlo en varias etapas. Durante la primera etapa los objetivos han sido:

- Comprender adecuadamente el software "R" y cómo utilizarlo para desempeñar las tareas de lectura y escritura de datos en ficheros.
- Estudiar el manejo de datos y los distintos formatos posibles.
- Utilizar algoritmos de modelado de datos.

Para poder experimentar y comprender dichos aspectos, hemos decidido utilizar una serie de datos de un proyecto anterior, del departamento de energética edificatoria, en el que se habían empleado ya técnicas de regresión lineal y múltiple. Dicho proyecto sigue el protocolo IPMVP-EVO (estándar de protocolos de medida y verificación energética) y consistía en refrigerar varias salas de control y seguridad de un túnel subterráneo. En concreto disponemos de las variables "temperatura en el pasillo" y "temperatura en la sala" para comprobar cómo afectan al consumo de energía de distintos compresores. De esta manera podemos ver el comportamiento de distintos algoritmos y aprender diversas técnicas del uso y optimización de los mismos.

A continuación, pasamos a explicar la primera etapa del proyecto.

3.1 Primera etapa

Esta etapa es fundamentalmente experimental y de lo que se trata es principalmente introducir el funcionamiento de los algoritmos que posteriormente se utilizarán para el modelado de datos y su estudio.

3.1.1 Algoritmo red neuronal

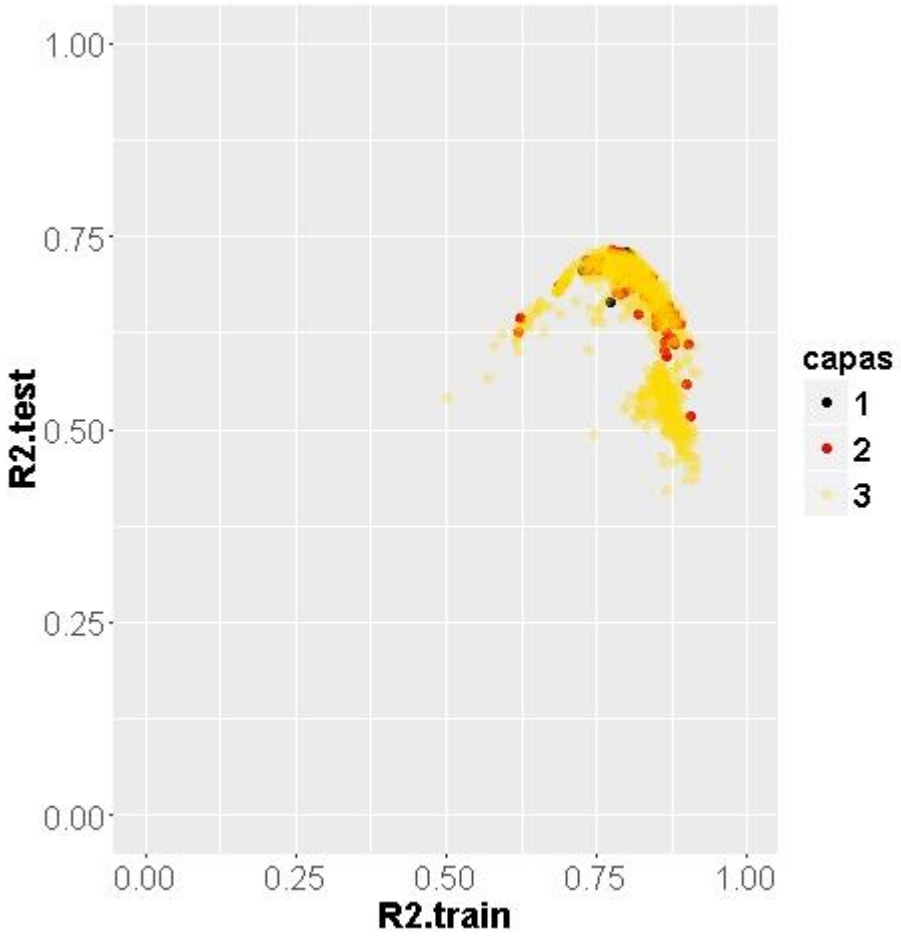
Para el entrenamiento y creación del modelo mediante un algoritmo de redes neuronales artificiales hemos decidido utilizar la libraría denominada "neuralnet". Primero se utilizan los ficheros "ejemplo_red_neuronal.r" y "ciclo_combinado.r" para comenzar a entender el funcionamiento de dicha librería así como la sintáxis del programa.

La librería "neuralnet" de R contiene una función, neuralnet(), que recibe varios argumentos que a continuación explicamos brevemente.

Parámetros:

- **Formula:** descripción simbólica de del modelo que se quiere ajustar. La estructura de una fórmula correcta sería: VPredecir ~ V1 + V2 + V3 … + Vn

- **Data:** una variable de tipo data.frame que contenga las varibles e instancias correspondientes a la fórmula especificada.
- **Hidden**: indica el número de capas ocultas así como el número de neuronas en cada una de dichas capas. La red de nuestro caso tendrá tres capas ocultas de 4, 3 y 3 neuronas respectivamente. No existe un algoritmo o método matemático que nos permita determinar con exactitud el número de capas ocultas y de neuronas en cada capa que se deben utilizar, sino que dependiendo de cada problema funcionará mejor con unos parámetros u otros, recayendo en el autor la tarea de elegir dichos parámetros según le convenga. Para llegar a la elección de las dimensiones de la red, hemos hecho una iteración de valores para las capas desde una sola capa hasta 3 capas con diez neuronas cada una de manera que nos quedamos con la configuración que mejores resultados obtenga. En la Gráfica 3 podemos ver el comportamiento del parámetro R^2 tanto en test como en train obtenidos tras las iteraciones mencionadas, de manera que nos permite elegir la configuración óptima de capas. Dicha configuración será aquella que tenga el valor de R en test por encima del 70% y la menor distancia al origen desde el valor R^2test + R^2train.



Gráfica 3: Comportamiento del parámetro R2 en train frente a test para redes neuronales de distintos tamaños.

- **Stepmax**: máximo número de iteraciones para el entrenamiento de la red. Si se alcanza este valor, el entrenamiento se detiene.

- **Rep**: número de repeticiones del entrenamiento de la red.
- **Algorithm**: indica el algoritmo principal en el que se basará la red; puede ser backprop (backpropagation), rprop+ (Resilient Backpropagation with weight backtracking), rprop- (Resilient Backpropagation without weight backtracking), sag o slr.
- **Otros**: existen más parámetros que se pueden utilizar, pero únicamente se explican estos ya que son los que se utilizan en el proyecto.

Esta función nos devuelve un objeto de clase "nn" con varias componentes: "net.results", que contiene los resultados obtenidos por la red, y "neurons", una lista de neuronas. Este objeto se utilizará en la función compute() junto con el conjunto de nuevos datos a predecir (test), y nos devolverá los resultados de los valores predichos.

3.1.2 Algoritmo regresión lineal

Para poder aplicar el algoritmo de regresión lineal, se utiliza la función "lm" de la librería "stats". La función recibe los parámetros fórmula y data como hemos visto ya, y devuelve un objeto de clase "lm" que tiene varias componentes;

- **Coefficients:** un vector de coeficientes.
- **Residuals:** los valores residuo o residuales, estos son, la respuesta menos los valores ajustados.
- **Fitted values:** los valores medios de ajuste.
- **weights:** los pesos utilizados (en caso de que los haya).

Hecho esto, el siguiente paso es la llamada a la función "predict.lm", a la cual se le pasa el objeto de clase "lm" obtenido anteriormente y los nuevos datos que se quieren predecir, y nos devuelve los valores predichos para la variable correspondiente de las nuevas instancias.

3.1.3 Algoritmo árbol de regresión

El algoritmo escogido para la utilización de un árbol de regresión es el proporcionado por la función rpart() de la librería con su mismo nombre. Esta función recibe cómo parámetros de entrada;

- **Formula:** descripción simbólica de del modelo que se quiere ajustar. La estructura de una fórmula correcta sería: VPredecir ~ V1 + V2 + V3 … + Vn
- **Data:** una variable de tipo data.frame que contenga las varibles e instancias correspondientes a la fórmula especificada.
- **Method:** acepta los valores "exp", para objetos de tipo 'survival', "poisson" para los que la variable a predecir tenga dos columnas, "class" para los valores no numéricos y "anova" para valores numéricos.

Devuelve un objeto de tipo "rpart" y éste se debe utilizar en la llamada a la función "predict", igual que como se hacía para la regresión lineal.

3.1.4 Algoritmo random forest

Por último, para el uso del algoritmo random forest, hemos elegido trabajar con la función "randomForest" de la librería con el mismo nombre. Esta función se basa en el algoritmo de *Breiman* y recibe como argumento de entrada los parámetros:

- **Formula:** descripción simbólica de del modelo que se quiere ajustar. La estructura de una fórmula correcta sería: VPredecir ~ V1 + V2 + V3 … + Vn
- **Data:** una variable de tipo "data.frame" que contenga las varibles e instancias correspondientes a la fórmula especificada.
- **Mtry:** número de variables que se elegirán aleatoriamente como candidatas en cada corte. Por defecto para regresión el valor es Nº variables/3.
- **Ntree:** número de árboles a utilizar. No debe ser un número pequeño para que se pueda asegurar que cada instancia se trata varias veces.
- **Importance:** Variable booleana que indica si se debe tener en cuenta la importancia de las variables según los predictores a la hora de seleccionar.

Se obtiene un objeto de tipo "randomForest" que se debe utilizar en la llamada a la función "predict", igual que como se hacía para la regresión lineal y para el árbol de regresión.

3.1.5 Modelización de datos

Una vez explicado brevemente el funcionamiento de las funciones y técnicas a utilizar, procedemos a la creación del modelo de datos. Como vimos anteriormente, los primeros pasos antes de aplicar técnicas de minería de datos implican la selección, análisis y transformación de los datos de entrada que se van a utilizar. A esta primera etapa se le denomina pre procesamiento de datos. El objetivo fundamental del pre proceso de datos es manipular y transformar los datos originales de tal forma que la información contenida en ellos pueda ser expuesta o facilitar el acceso a ella. La preparación de los datos genera datos de calidad que pueden conducirnos a patrones y reglas de mayor calidad más fácilmente. El pre procesamiento de datos engloba o se compone de varias tareas o etapas:

- **Colección e integración de datos:** proceso de integrar los datos provenientes de diferentes fuentes. Se resuelven problemas de representación y codificación mediante la integración de datos de diferentes tablas para crear información homogénea. Se deben tener en cuenta posibles errores de medida.
- **Transformación de datos:** normalización y agregación. Consiste en transformar los valores de tal forma que todos los atributos estén en el mismo rango. Existen varias técnicas, en este proyecto se verán tres distintas; máximo y mínimo, estándar y máximo y mínimo centralizados.
- **Limpieza de datos:** Proceso de eliminar los errores e inconsistencias de los datos y resolver el problema de la identificación de objetos. Incluye el tratamiento de valores con ruido, identificación de *outliers* y completar o eliminar valores perdidos debidos a errores técnicos (de equipamiento), datos no introducidos, etc.
- **Discretización de datos:** modificar el tipo de datos, normalmente aplicado a atributos numéricos. Transformar los valores numéricos en discretos o viceversa. Existen

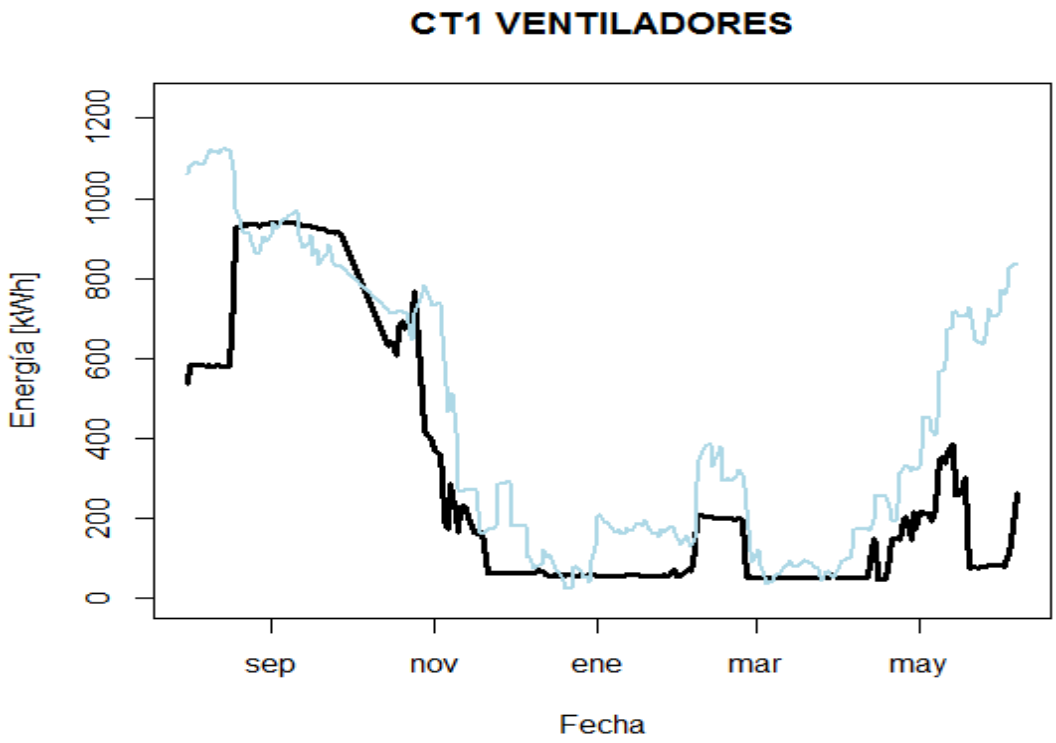
técnicas de minería de datos que solo aceptan atributos discretos por lo que esta tarea puede ser esencial. En nuestro caso, no se llevará a cabo ya que se trabaja con los valores numéricos originales o normalizados, pero en cualquier caso no es necesario que sean discretos.

- **Reducción de datos:** Obtener representación reducida en volumen, pero produce resultados analíticos iguales o similares.

Por tanto, después de ver los pasos del pre procesamiento de datos, podemos comenzar a llevarlos a cabo. El primer paso consiste en leer los datos de cada fichero y agregarlos de manera que los tengamos en una única variable. Los datos de esta primera etapa son fácilmente agrupados ya que únicamente están compuestos por tres variables (consumo a predecir, temperatura exterior y temperatura interior). Lo siguiente es seleccionar el porcentaje de ejemplos empleados para el entrenamiento de cada modelo y para el conjunto de validación. Así, lo que se hemos hecho ha sido probar con un conjunto de entrenamiento formado por el 70% y con otro formado por el 80% de los datos.

Los resultados obtenidos son los siguientes:

Con el 80% de los datos para el conjunto de entrenamiento.

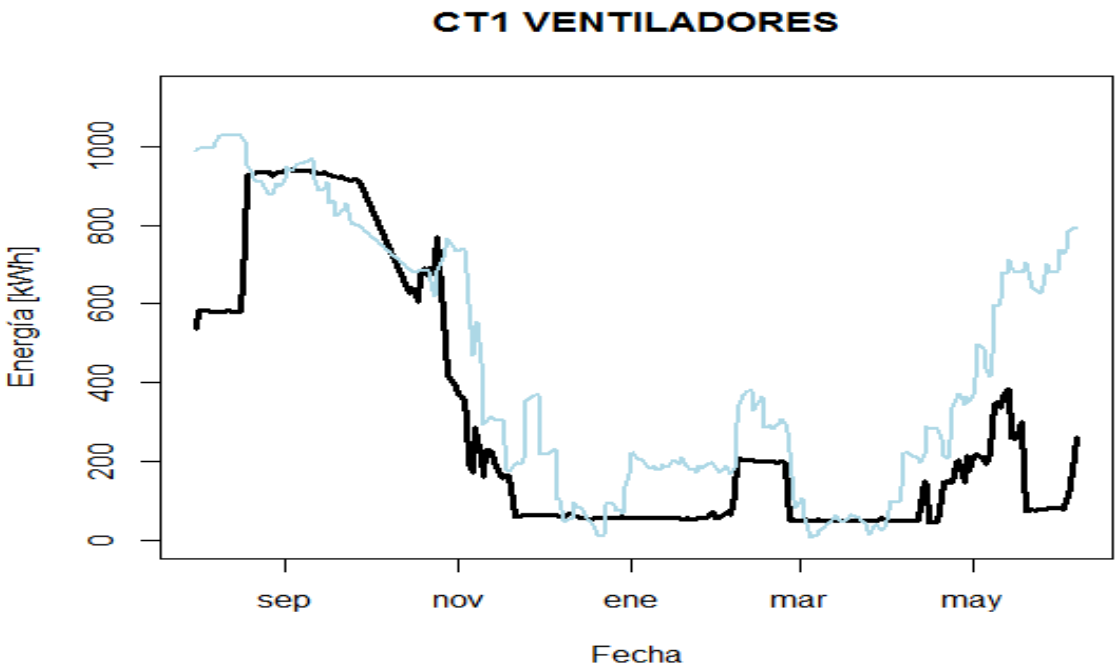


Gráfica 4: Resultados obtenidos mediante el uso de una red neuronal que utiliza el 80% de los datos para entrenar y el 20% restante para realizar la predicción.

Train: MSE =1127.885 RMSE = 33.584

Test: MSE = 207621.819 RMSE = 455.655

Con el 70% de los datos para el conjunto de entrenamiento.



Gráfica 5: Resultados obtenidos mediante el uso de una red neuronal que utiliza el 70% de los datos para entrenar y el 30% restante para realizar la predicción.

Train: MSE = 1311.89 RMSE = 36.49

Test: MSE = 231106.14 RMSE = 480.735

Debido a que el error cuadrático medio en *test* es menor (aunque sea alto) utilizando el 70%, independientemente del método de normalización utilizado, será ese porcentaje el utilizado de aquí en adelante.

Una vez elegido el porcentaje de valores a utilizar para cada conjunto, lo siguiente que hacemos es probar utilizando la técnica de *Bootstrapping*.

Partiendo de una muestra de datos para los cuales se calcula un estadístico de interés (por ejemplo, una media o un coeficiente de correlación), el método consiste en:

- 1) Crear un gran número de sub-muestras con reposición de los mismos datos, por ejemplo, 2000 muestras.

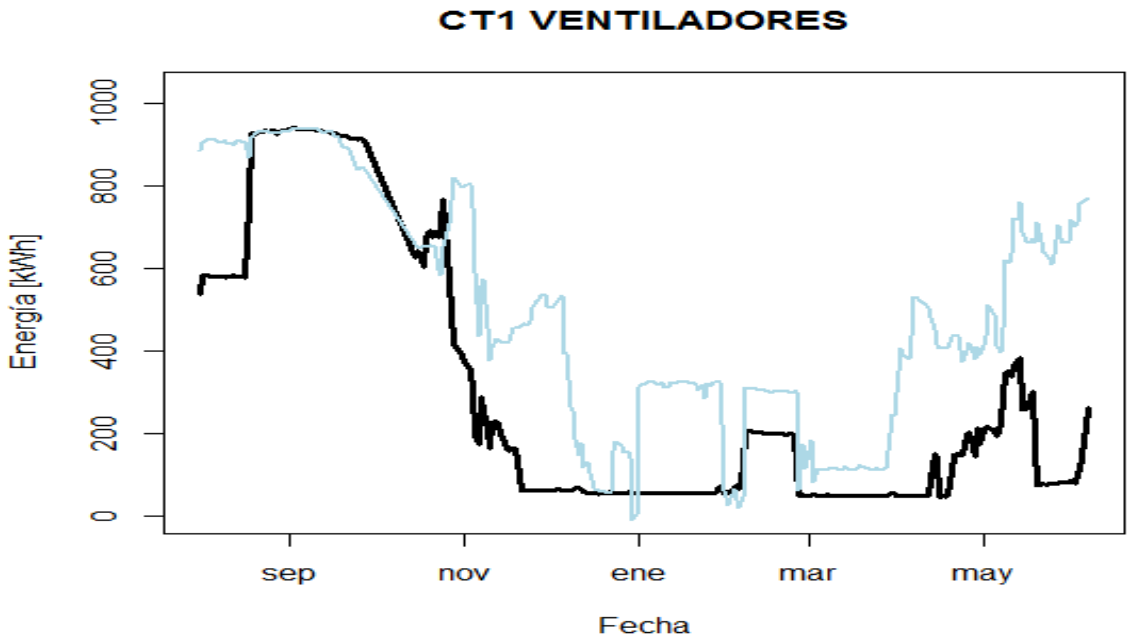
- 2) Calcular para cada muestra resultante el valor del estadístico en cuestión. Se obtiene así una aproximación a la distribución de muestreo del estadístico, a partir de la cual podemos construir un intervalo de confianza para dicho estadístico o realizar una prueba de significación.

Se puede prever que los resultados no serán muy buenos, ya que los datos son muy variables y no nos sirve con la media.

En el proyecto hemos utilizado dos variantes a la hora de aplicar el *bootsrap*; una en la que se crean varios conjuntos con reemplazo aleatorio y después se construye un conjunto final con

la media de cada uno de los anteriores, y otra en la que se escoge aleatoriamente uno de los subconjuntos creados y se utiliza directamente para entrenar y validar. Los resultados son:

Conjunto con reemplazo:

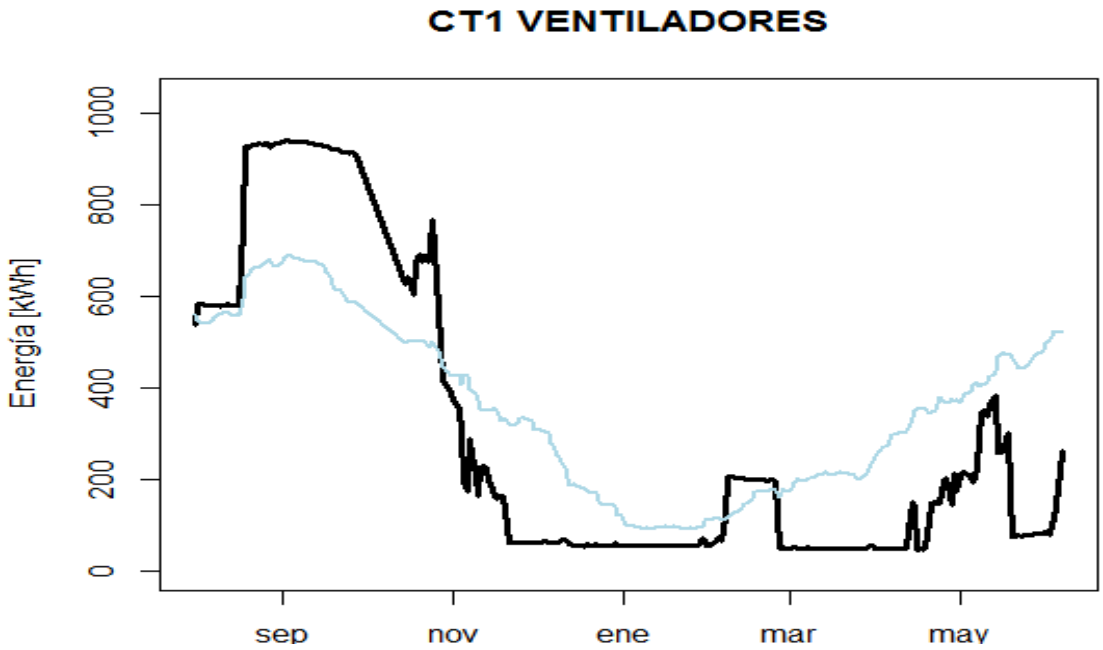


Gráfica 6: Resultados obtenidos mediante el uso de una red neuronal que entrena y testea con un subconjunto de datos obtenido mediante reemplazamiento de boostrapping.

Train: MSE = 1911,896 RMSE = 43.725

Test: MSE = 363767.003 RMSE = 603.131

Conjunto de medias:



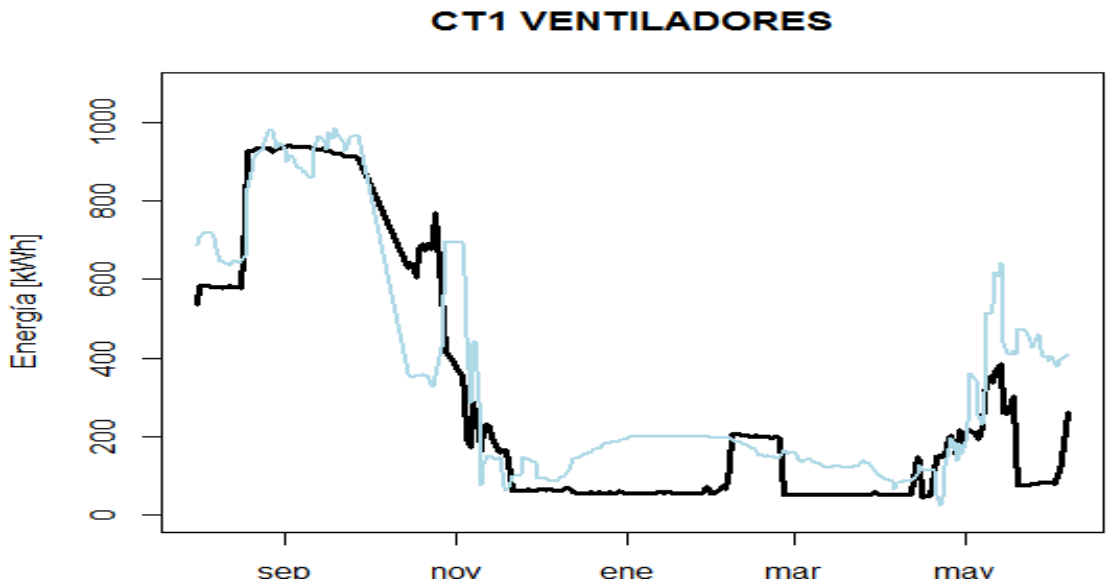
Gráfica 7: Resultados obtenidos mediante el uso de una red neuronal que entrena y testea con un subconjunto de datos obtenido mediante la media de los conjuntos creados por boostrapping.

Train: MSE = 8972.826 RMSE = 94.725

Test: MSE = 784035.870 RMSE = 885.458

Lo siguiente que hemos probado ha sido una división de los datos (clusterización) en sectores "manualmente". Esto es, se dividen los datos de forma que tenemos: ejemplos que cumplan tpasillo < 15º, otros que cumplan >= 15 & tpasillo <= 25 y un tercer sector tal que tpasillo > 25. La elección de esta división manual se debe a los resultados obtenidos en simulaciones anteriores (recordamos que en esta primera etapa utilizamos datos de un proyecto ya terminado) al momento de realizar este proyecto.

Cada uno de los sectores o clústeres representa un conjunto que datos que se evaluarán por separado, es decir, se utilizarán tres redes neuronales distintas de forma que se aprenda cada uno de los grupos. Una vez hecho esto, se combinan los resultados y se representan de forma gráfica. Los resultados obtenidos son:



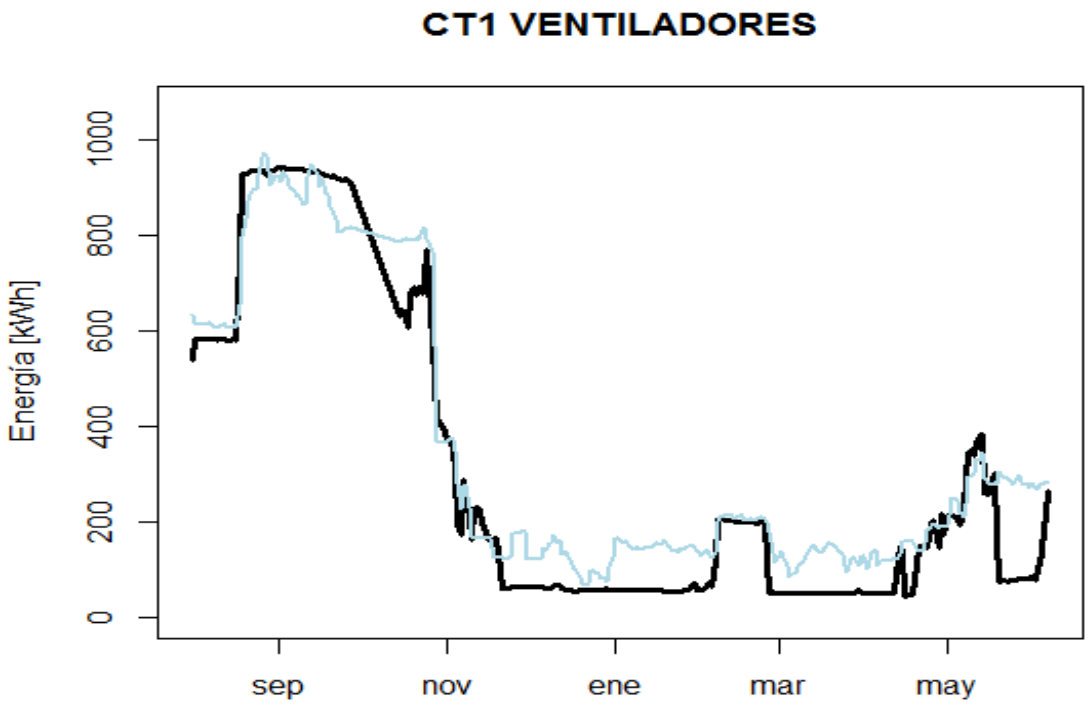
Gráfica 8: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto de datos divididos manualmente como se ha visto anteriormente.

Train: MSE = 1697.553 RMSE = 41.201

Test: MSE = 269886.484 RMSE = 519.506

Después hacemos una división en clústeres o sectores pero no manual, sino utilizando técnicas de clusterización para obtener el número óptimo de clúster (que coincide en que son 3, pero no se dividen de la misma que forma que antes). Una vez obtenido el número óptimo de clústeres se utiliza el algoritmo *Kmeans* para separar los datos y asignarles un clúster.

Cuando ya tenemos todos los datos asignados a alguno de los clústeres, se procede de igual forma que antes, es decir, se utiliza una red neuronal para cada sector y después se unen los resultados y se representan. Dichos resultados son los siguientes:

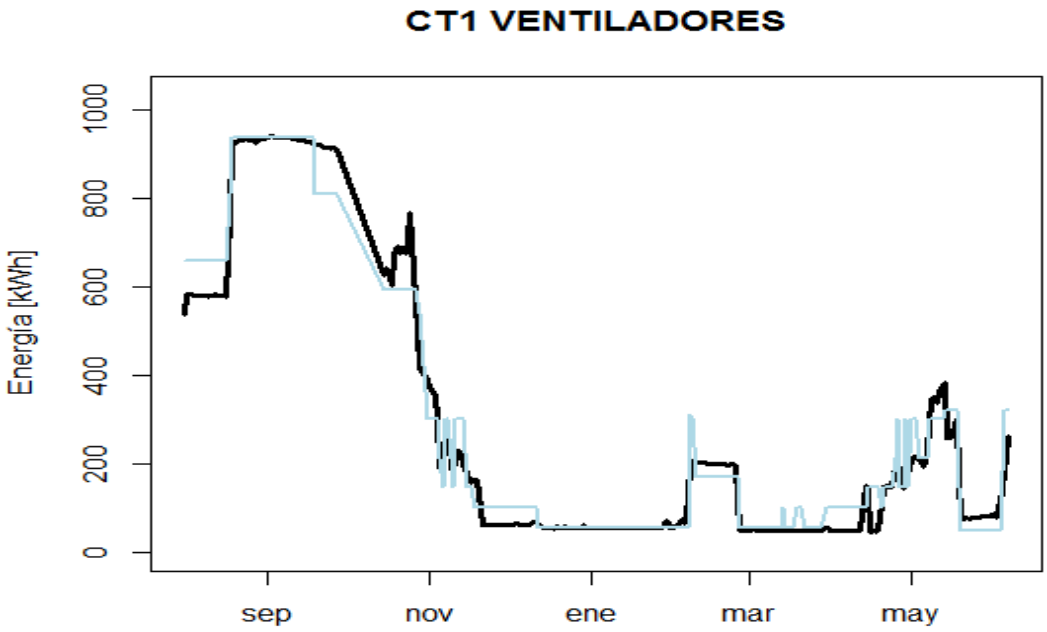


Gráfica 9: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto de datos divididos mediante la técnica K-means.

Train: MSE = 29.44 RMSE = 5.426

Test: MSE = 6716.997 RMSE = 81.957

Lo siguiente que hemos utilizado ha sido la función BDK o algoritmo de Kohonen. Esta técnica consiste en crear un mapa auto-organizado. Los mapas auto-organizados son un tipo de red neuronal artificial que es entrenada usando aprendizaje no supervisado para producir una representación discreta del espacio de las muestras de entrada, llamado mapa. Los mapas auto-organizados son diferentes de otras redes neurales artificiales, en el sentido que estos usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada. Los resultados obtenidos utilizando esta función son:



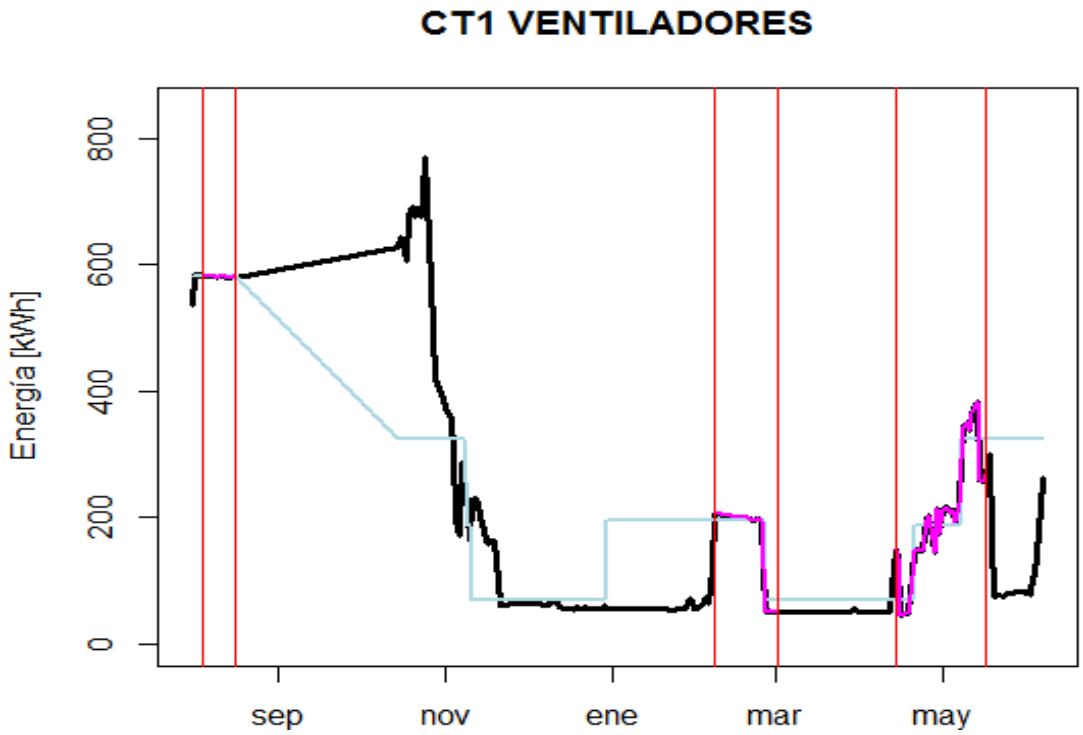
Gráfica 10: Resultados obtenidos mediante el uso de un algoritmo de mapas auto-organizados de Kohonen.

Train: MSE = 330.8 RMSE = 18.188

Test: MSE = 13000 RMSE = 114.02

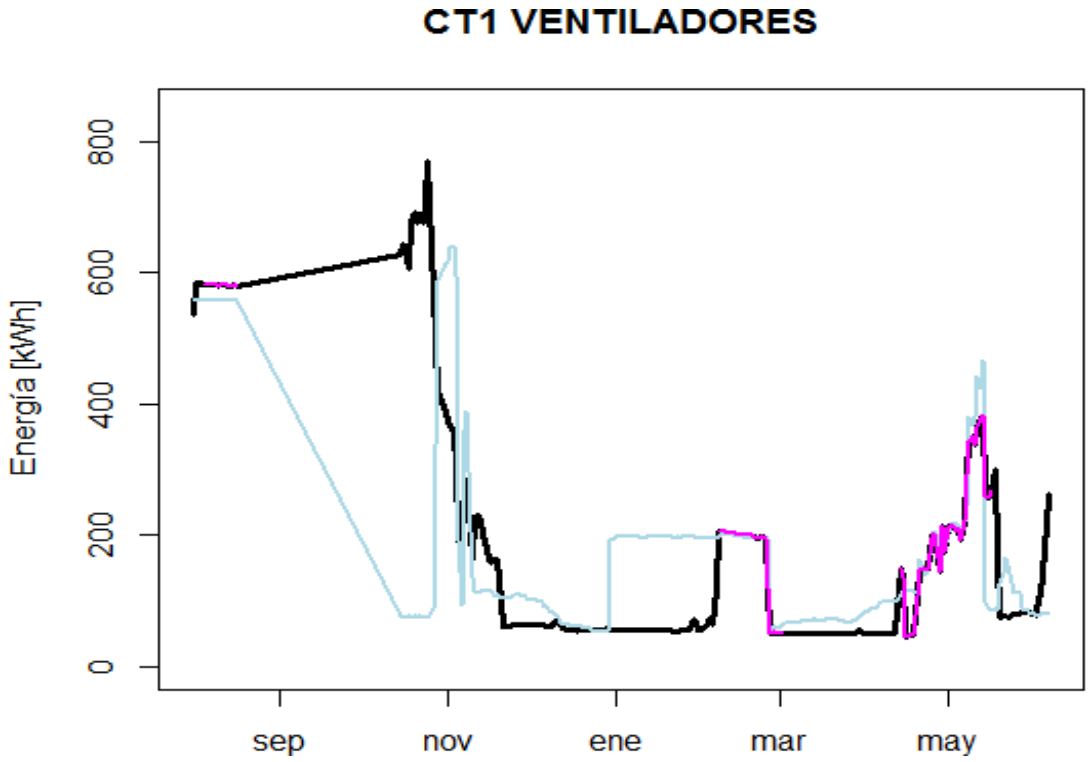
La función utilizada en nuestro caso para trabajar con árboles es la denominada rpart(). Simplemente se introducen los datos de entrenamiento (no hace falta que estén normalizados ni centralizados) y un parámetro que indica que el árbol que se pretende crear es de regresión (también puede ser de clasificación si los datos son discretos).

En la figura siguiente se muestran los resultados obtenidos mediante el uso de esta técnica. Se puede apreciar una forma de escalera. Esto se debe a que esta técnica distingue una serie de características que influyen en los datos de manera que es capaz de generar grupos o clusters de datos similares entre sí. Para cada uno de los grupos realiza la media, de forma que cuando llega un nuevo dato, sigue los nodos y hojas creados por el camino correspondiente según los valores obtenidos y al final es asignado a uno de los grupos y por tanto a una de las medias.



Gráfica 11: Resultados obtenidos mediante el uso de un algoritmo de árbol de decisión.

En la Gráfica 11 los datos que aparecen entre líneas verticales rojas son aquellos en los que el sistema de frío estaba parado y que se utilizan para testear. El siguiente paso es utilizar las divisiones y reglas obtenidas del árbol de manera que podamos aplicar la técnica de RRNN vista anteriormente pero con unos nuevos grupos de los mismos datos.



Gráfica 12: Resultados obtenidos mediante el uso de una red neuronal para cada subconjunto determinado por la técnica de árbol de decisión anterior.

En los casos en los que se tienen muchas variables de entrada, se suelen aplicar técnicas de selección de variables para optimizar los cálculos y métodos. Uno de los algoritmos más conocidos es el PCA (Análisis de Componentes Principales). En nuestro caso se puede llevar a cabo este estudio pero no merece la pena realmente ya que únicamente estamos utilizando dos variables. Si utilizamos el PCA podemos observar que la variable de más peso es la de Tª pasillo, pero cómo decíamos, al tener solo dos variables no nos vamos a quedar únicamente con la de más valor. Esta técnica, PCA, la tendremos en cuenta en la segunda etapa del proyecto en la que se trabaja con un mayor número de variables.

A continuación aplicaremos estas técnicas pero con otros datos. Hasta ahora esto eran pruebas experimentales con datos de un proyecto distinto al del objeto de estudio.

3.2 Segunda Etapa

3.2.1 Supermercado Ali-Gobeo

Después de estudiar varias posibilidades y opciones con datos reales de un proyecto a modo de entrenamiento, la segunda parte del proyecto consiste en aplicar los conocimientos adquiridos a los datos del proyecto objeto de estudio. Para entender el objetivo del proyecto es necesario explicar el motivo del mismo;

La compañía EROSKI inauguró un establecimiento con un innovador sistema de trigeneración capaz de suministrar electricidad, calor y frío a partir de biomasa. Además de autoabastecerse de fuentes renovables de energía que reducirían drásticamente el consumo proveniente de la red eléctrica, se instalaron diversas medidas de ahorro energético. Ahí es donde entra CENER; una de las tareas consistía en la realización de una evaluación de impacto sobre el sistema eléctrico español y la reducción del consumo de energía lo máximo posible.

Pues bien, la tarea que nos ocupa ahora es la de realizar un estudio de los datos de consumo obtenidos tras la aplicación de las diversas medidas de ahorro energético instaladas mediante el uso de distintas técnicas y algoritmos de estimación y modelización de datos.

El supermercado elegido por EROSKI para la implementación del proyecto europeo ZERO STORE, es un EROSKI CENTER de 1433 m^2 situado en la ciudad de Vitoria-Gasteiz (Green Capital 2012) en el polígono industrial de Ali-Gobeo. Al hilo de la elección del supermercado, una de las tareas también desarrolladas ha sido la creación de una herramienta simple que permitirá a EROSKI comparar los distintos establecimientos disponibles y analizar cuáles de ellos son los más propicios a ser reestructurados tal y como se realizó en el proyecto del supermercado mencionado. El funcionamiento y descripción de esta herramienta se puede ver en el apartado ANEXO 2.

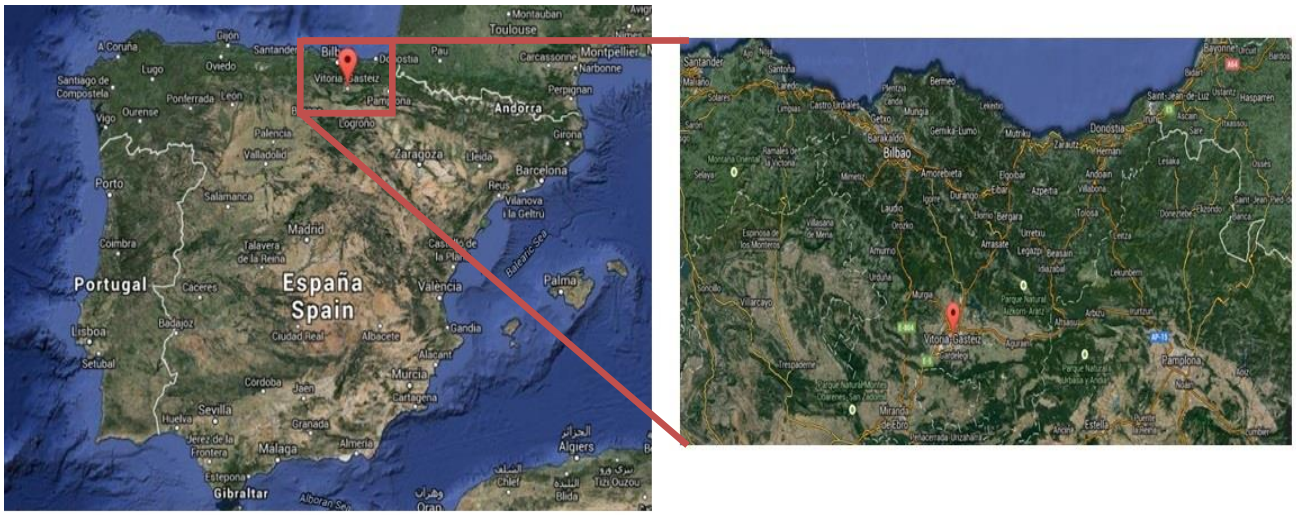


Ilustración 1: Localización de la ciudad de Vitoria en un mapa de la península ibérica.

El supermercado tiene acceso desde dos calles, la entrada principal y aparcamiento desde la calle Bremen lindando con una zona residencial y el acceso trasero para reposiciones desde la calle Stuttgart, que forma parte de un polígono industrial. La zona climática en la que se encuentra el establecimiento es la zona D1 (un poco más adelante se explicarán las características de la zona climática correspondiente, D1).

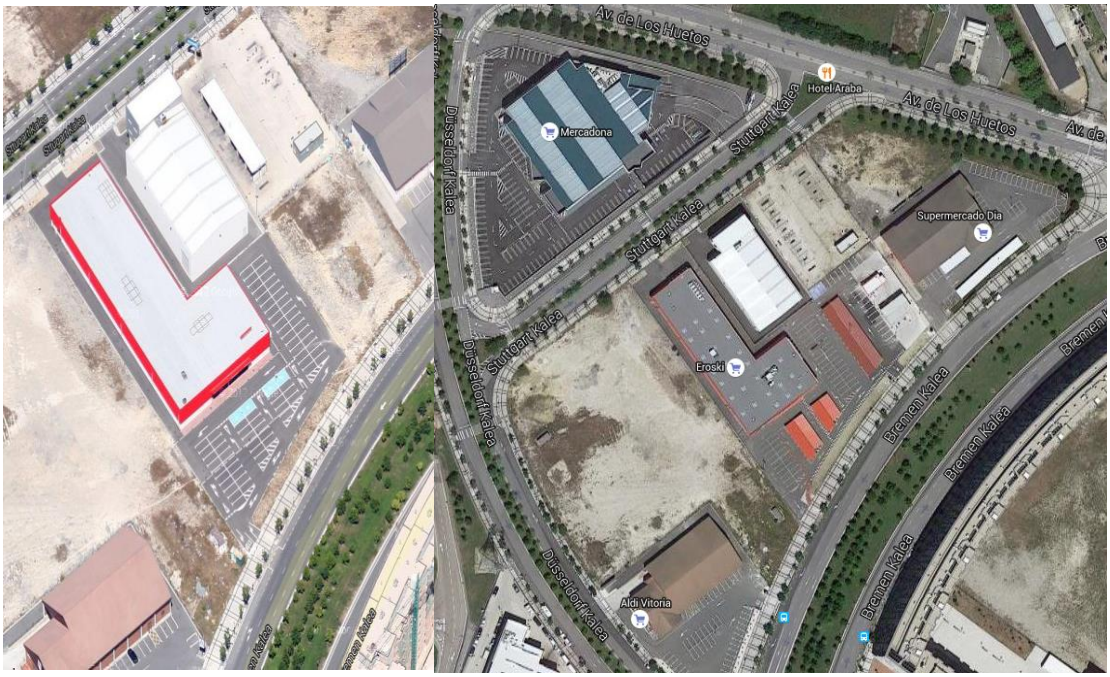


Ilustración 2: Vista aérea del supermercado de Eroski.

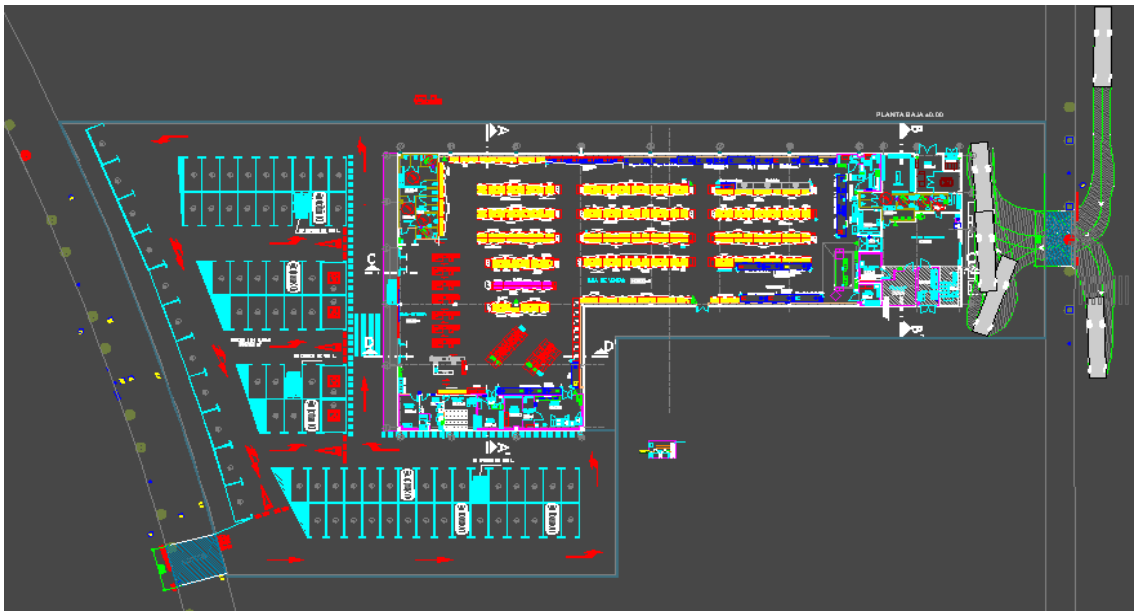


Ilustración 3. Plano del supermercado

La tienda abre de lunes a sábado en horario de 9:00 a 21:00. Se ha realizado la obra y la apertura de una gasolinera en las inmediaciones del supermercado. Se trata de una gasolinera que se halla también bajo la dirección del grupo EROSKI.

Tenemos datos reales del consumo eléctrico del supermercado, obtenidos gracias a 27 analizadores de red, que nos permiten caracterizar el consumo eléctrico que supone cada uso del supermercado. Un analizador de red puede medir una gran variedad de parámetros eléctricos. Se trata de unos analizadores de la compañía CIRCUTOR. La monitorización se ha llevado a cabo desde el año 2014.

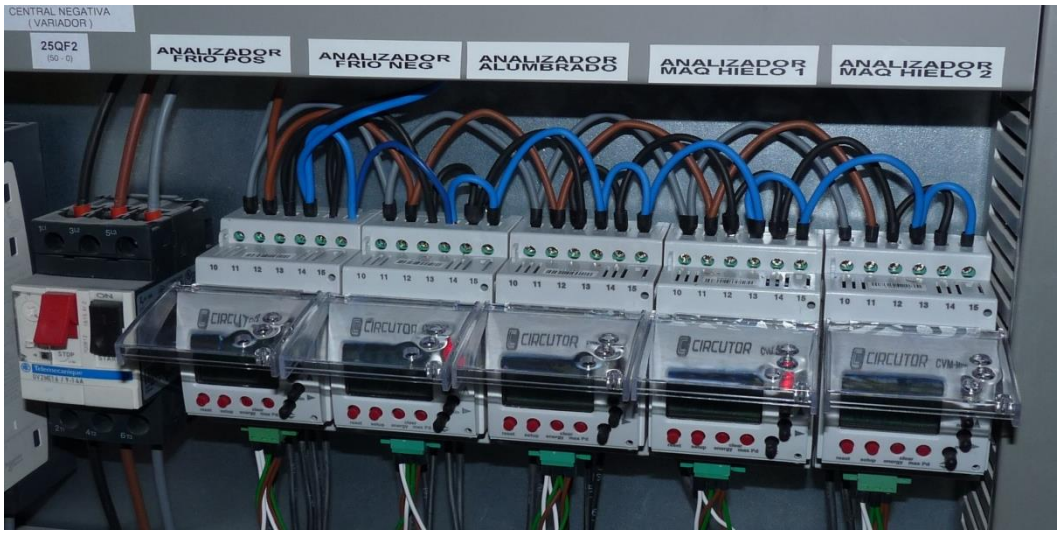


Ilustración 4. Los 5 analizadores colocados en el cuadro secundario de frío.

Los analizadores de red son de dos modelos diferentes, unos son del modelo ‘CVM-NRG96’ y otros del modelo ‘CVM-MINI’.

Los 27 analizadores de red que están en el supermercado reciben la siguiente denominación:

1. Cuadro General
2. Alumbrado Exterior 1
3. Alumbrado Exterior 2
4. Alumbrado Sala de Ventas 1
5. Alumbrado Sala de Ventas 2
6. Alumbrado Sala de Ventas 3
7. Cortina 1
8. Cortina 2
9. Cuadro Secundario de Frescos (Fuerza + Alumbrado)
10. Frescos Preferente (Alumbrado)
11. Cuadro Secundario de Frío
12. Frío 1
13. Frío 2
14. Frío 3
15. Frío 4
16. Frío 5
17. Panadería no Preferente (Fuerza)
18. Horno 1
19. Horno 2
20. Panadería Preferente (Alumbrado)
21. Roof Top 1
22. Roof Top 2
23. Cuadro secundario de Oficinas
24. SAI Oficinas
25. Vestuarios (Alumbrado)
26. Almacén (Fuerza)
27. Sala de Ventas (Fuerza)

Para facilitar el análisis de los datos se agruparon los consumos registrados por los analizadores de la siguiente manera:

1. **Cuadro General**: el analizador número 1 (‘Cuadro General’).
2. **Equipos de Frío:** el analizador número 11 (Suma de ‘Fríos’ 1-5 más los consumos no registrados).
3. **Climatización:** la suma de los consumos de las dos Cortinas y las dos ‘Roof Top’.
4. **Alumbrado:** suma del consumo del alumbrado de la superficie de ventas, exterior, panadería y vestuarios.
5. **Fuerza:** suma de los consumos de fuerza de la sala de ventas, almacén y panadería.
6. **Frescos:** el analizador número 9 ‘Cuadro Secundario de Frescos (Fuerza + Alumbrado)’.
7. **Oficinas:** representa el consumo total de las oficinas del supermercado. Calculado como la suma de los analizadores 23 y 24 (‘Cuadro secundario de Oficinas’ + ‘SAI Oficinas’).
8. **Otros:** resto de consumos y los consumos no registrados por ningún analizador específico pero sí por el analizador ‘Cuadro General’ (1).

También se dispone de sensores de temperatura y humedad que proporcionan datos tanto del interior como del exterior del establecimiento.

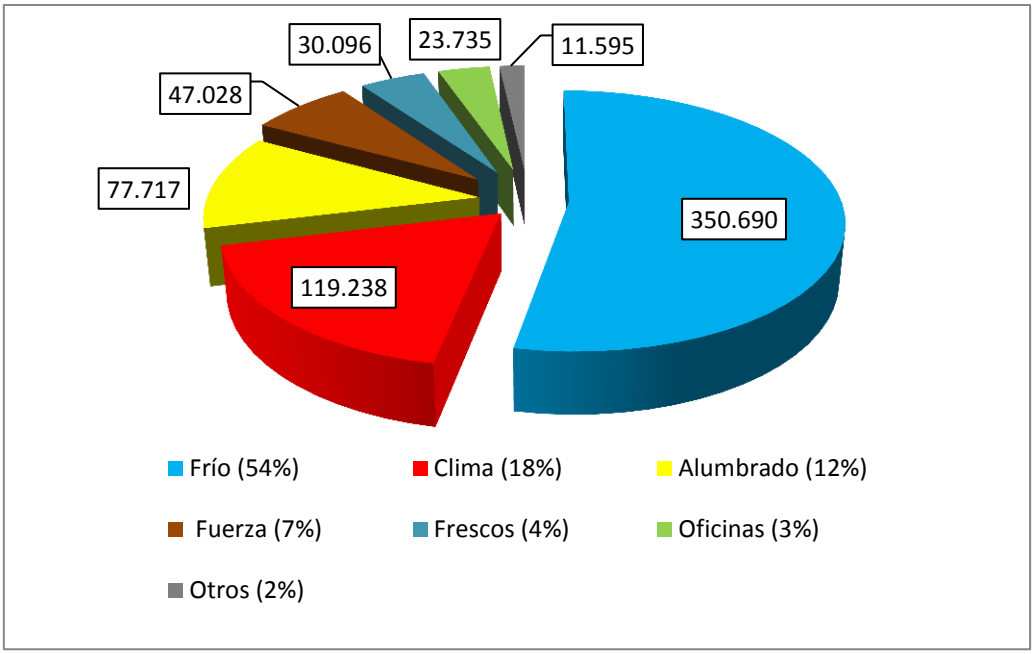
Además disponemos también de las mediciones oficiales de la temperatura en el exterior (Vitoria), gracias a EUSKALMET.

Se colocaron varios componentes tanto dentro como fuera del supermercado para medir la temperatura.

Ilustración 5: Analizador de red portátiles.

Ilustración 6: Medidor de temperatura portátil colocado en el CPD de CENER del mismo modo que se colocaron en el supermercado.

Agrupando convenientemente, en siete grupos, los datos de consumo descargados ha sido posible realizar un Breakdown para el año 2014, el cual está representado en la Gráfica 13.

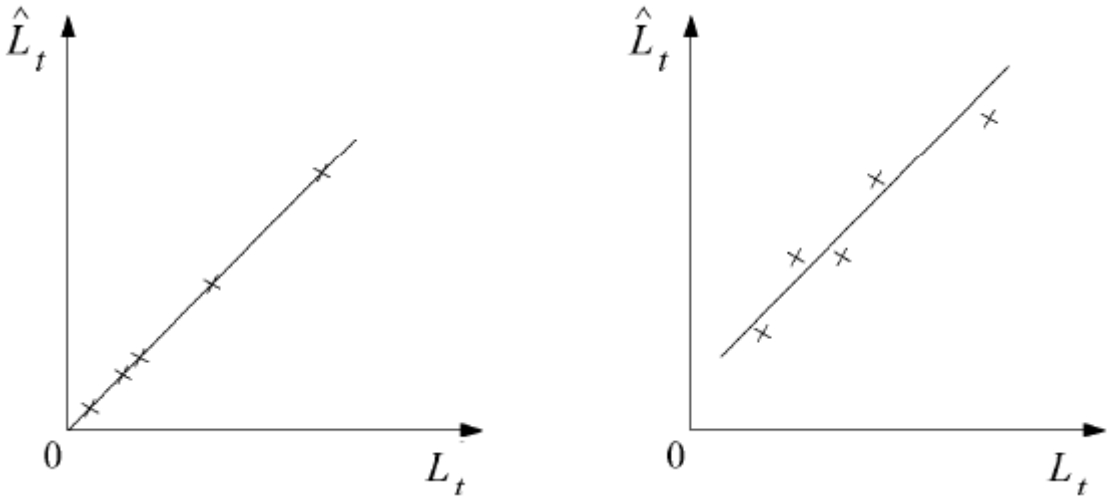


Gráfica 13: Consumo total por sectores.

El consumo del cuadro secundario de frío representa el 54% del consumo total, lo que asegura que este consumo es el más importante.

Históricamente la energía eléctrica se ha considerado un bien indispensable y de gran interés estratégico para la economía de los países avanzados. Las compañías eléctricas y los grandes consumidores adquieren la energía en función de sus previsiones de consumo, y en la medida que pueden estimar con precisión ese gasto, pueden ajustar los volúmenes de compra por lo que poder conseguir predecir la demanda de energía correctamente y con la antelación suficiente es una gran ventaja. Incluso para empresas que no compren directamente la energía en función del consumo, conseguir una buena predicción puede ser de gran ayuda para afrontar nuevas medidas de ahorro, realizar estudios y análisis del consumo, etc. Una predicción inferior al consumo provoca que se deba comprar más energía por medios más caros de los que se hubieran utilizado con una predicción correcta, lo que supone lógicamente pérdidas para la empresa. Del mismo modo, sobreestimar la demanda también supone pérdidas ya que la energía sobrante no se aprovecha cómo debería. Por tanto, la predicción de la demanda de energía con el menor error posible ha sido de gran interés por múltiples motivos como la mejora del funcionamiento de los mercados, para mejorar el servicio proporcionado por las empresas eléctricas a sus clientes, facilitar la gestión del sistema eléctrico, etc. Sin embargo, el consumo de energía depende de muchos factores como la temperatura, el día de la semana, el tipo de día, etc., lo que implica que su predicción sea un problema complejo que requiere el uso de técnicas sofisticadas.
Desde el punto de vista de la respuesta de la demanda es muy importante, y actualmente constituye una barrera en la implantación de los distintos programas, saber cuál es la predicción del consumo de un cliente para un determinado día, de forma que al modificar su consumo como respuesta a una señal del administrador del programa, con el uso de ésta, sea posible evaluar la acción tomada y sirva de base para realizar una correcta y justa retribución.
Una forma de llevar a cabo la valoración de la capacidad predictiva de los modelos es a través de la comparación entre las predicciones y los valores de consumo reales. Dado que los métodos de estimación suelen minimizar sumas de cuadrados de los residuos dentro de los períodos muestrales, la valoración de los modelos debe hacerse con datos diferentes a los utilizados para estimar. Una práctica frecuente consiste en dividir los datos muestrales en dos

subconjuntos. El primero se utiliza para estimar y el segundo para comparar las predicciones del modelo con las observaciones.
Una de las técnicas utilizadas es la obtención de los parámetros de la recta de regresión.
En el caso ideal de que la predicción fuera perfecta, la recta de regresión debería coincidir con la bisectriz del primer cuadrante del diagrama de predicciones-realizaciones.



Gráfica 14: Diagrama de predicciones-realizaciones

Mucho más común, en la literatura, es el cálculo de un parámetro que cuantifica la diferencia entre el consumo estimado y el real. Existen gran cantidad de índices, de entre los que se destacan:

Error cuadrático medio (ECM)
El error cuadrático medio de las predicciones se define como la media aritmética de los cuadrados de los errores de predicción.

$$ECM = \frac{\sum_{t=1}^{N}(\hat{L}_t - L_t)^2}{N}$$

Donde N es el número de datos estimados, $L\hat{}_t$ es el consumo estimado para el instante t y L_t es el consumo real para dicho instante.

Raíz cuadrada del error cuadrático medio (RECM)
El error estándar de las predicciones es la raíz cuadrada del error cuadrático medio (RECM):

$$RECM = \sqrt{\frac{\sum_{t=1}^{N}(\hat{L}_t - L_t)^2}{N}}$$

Estos dos índices son muy utilizados, aunque presentan un inconveniente y es que vienen expresados en las mismas unidades de medida que Lt , lo que implica que sus valores dependen en cada caso de la unidad de medida adoptada y no sirven para llevar a cabo comparaciones entre distintos métodos si se han tomado unidades distintas. Este defecto puede subsanarse sin dificultad, por ejemplo, dividiendo el índice por la media o por la desviación estándar.

Mean Absolute Percentage Error (MAPE)

Este error es el más utilizado en las diferentes publicaciones actuales y es el Mean Absolute Percentage Error (MAPE), que se define como:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left|\hat{L}_t - L_t\right|}{L_t}\times 100$$

Al ser un índice relativo, no presenta unidades, y permite comparar resultados de distintos modelos, incluso aplicados a distintos puntos de consumo.

Error medio de Energía (EME)

En el caso de que L_t represente el consumo de energía en un determinado periodo de tiempo, es posible definir el Error Medio de Energía (EME), cuya expresión es:

$$EME = \sum_{t=1}^{N}\frac{\left|\hat{L}_t - L_t\right|}{\sum_{t=1}^{N} L_t}\times 100$$

Muy similar al anterior, con la diferencia de que es menos sensible que el MAPE en aquellos casos en los que el valor real del consumo es inferior a la unidad.

Anteriormente hemos comentado que en el cálculo de la previsión del consumo eléctrico de un determinado sistema es posible observar que depende, en gran medida, de la actividad que haya en dicho sistema. Así, según la fecha para la que se realiza el cálculo de la estimación se debe disponer de un parámetro que indique la actividad prevista en el mismo. En la metodología utilizada por REE (Red Eléctrica Española) se establecen unos criterios para asignar dicho parámetro, llamado laboralidad, que son:

- El factor toma valores de 0,6429 hasta 1,0000.
- El día 1 de enero presenta el valor más bajo del año.
- El valor para los sábados dentro de un mismo mes es el mismo. Lo mismo se cumple para los domingos.
- Los martes, miércoles y jueves, que no sean festivos, habitualmente presentan un valor máximo.
- En los días festivos los valores para cada año cambian según el día de la semana.
- En un determinado día que sea festivo en una comunidad entonces el parámetro adquiere un valor inferior al de un día habitual.
- En los meses de julio y agosto no existen días con valor igual a 1,0000.
- Todos los años tienen los mismos valores con las modificaciones particulares de los festivos según el día de la semana.

En el presente trabajo se utiliza también el parámetro de laboralidad basado en los aspectos presentados.

Así pues, se procede a explicar los pasos ejecutados para la aplicación de las diversas técnicas a los datos objeto de estudio. Conviene recordar rápidamente los pasos o etapas que se deben seguir:

- **Integración y recopilación:** Comprensión del dominio de aplicación del problema, identificación de conocimiento a priori y creación del conjunto de datos.
- **Pre procesamiento:** Selección de datos, limpieza, reducción y transformación.
- **Selección de la técnica:** aplicación de algoritmos concretos de aprendizaje.
- **Evaluación:** interpretación y presentación de los resultados obtenidos.
- **Difusión:** utilización del nuevo conocimiento.

3.2.2 Datos y parámetros

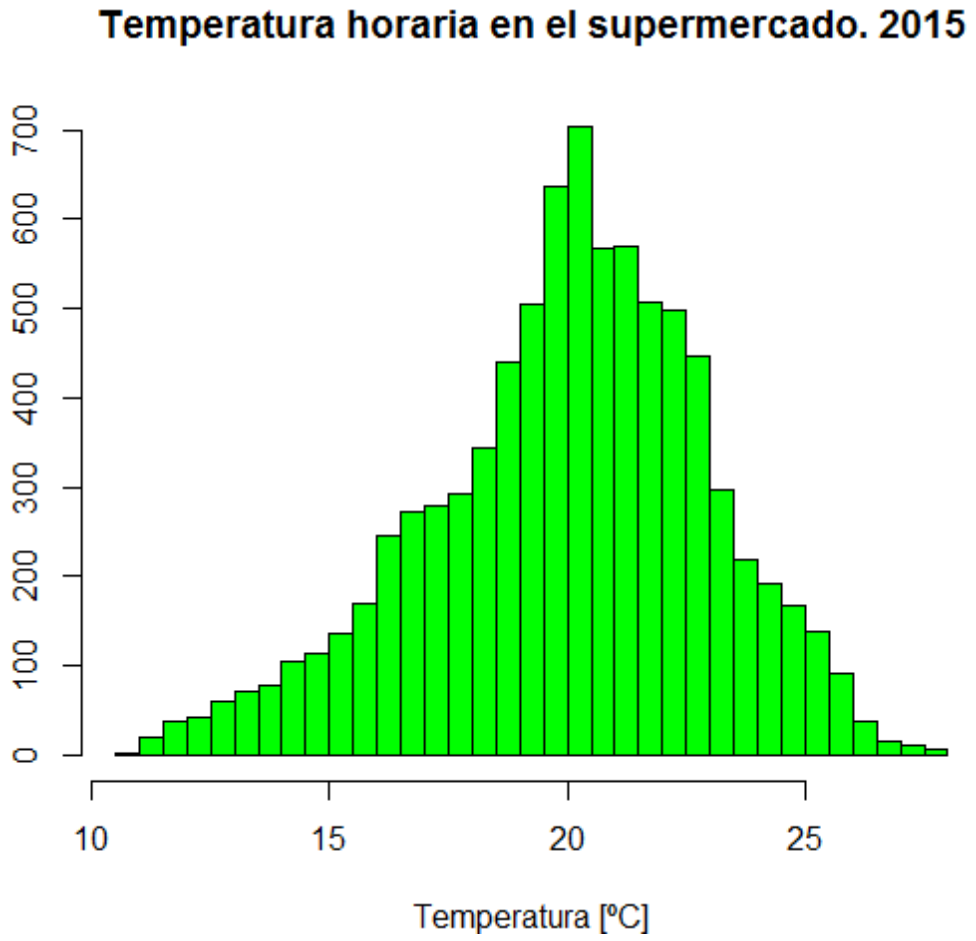
El primer paso es la selección del conjunto de datos y parámetros a utilizar. Tenemos varias variables que pueden influir en los resultados y se deben estudiar. Así pues, los datos que utilizaremos son:

- Archivo “temphrint2015.csv”: Contiene información relativa al interior del supermercado. Tiene las siguientes variables:
  - Fecha y hora: fecha y hora de la medición correspondiente.
  - Humedad relativa (%).
  - Humedad absoluta (g/m3)
  - Temperatura (Cº)
  - Punto de rocío (Cº)
- Archivo “temphrext2015.csv”: Contiene información relativa al exterior del supermercado. Tiene las siguientes variables:
  - Fecha y hora: fecha y hora de la medición correspondiente.
  - Humedad relativa (%).
  - Humedad absoluta (g/m3)
  - Temperatura (Cº)
  - Punto de rocío (Cº)
- Archivos “alumbradosv12015.csv", “alumbradosv22015.csv" y “alumbradosv32015.csv": Contienen información relativa al consumo energético del supermercado debido al alumbrado. Tiene las variables:
  - Fecha y hora
  - Consumo (kWh)
- Archivos "frio12015.csv" y “frio22015.csv”: Contienen la información del consumo energético del supermercado debido al frío positivo y negativo respectivamente. Tienen las variables:
  - Fecha y hora
  - Consumo (kWh)
- Archivos "cortina12015.csv" y "cortina22015.csv": Contiene la información del consumo energético de las cortinas.

- Archivos "rooftop12015.csv" y "rooftop22015.csv": Contiene la información del consumo energético del tejado.
- Archivo "cgbt2015.csv": Contiene los datos del consumo general de la tienda (Cuadro general baja tensión). Esta es la variable que queremos predecir. Tiene;
    - Fecha y hora
    - Consumo (kWh)

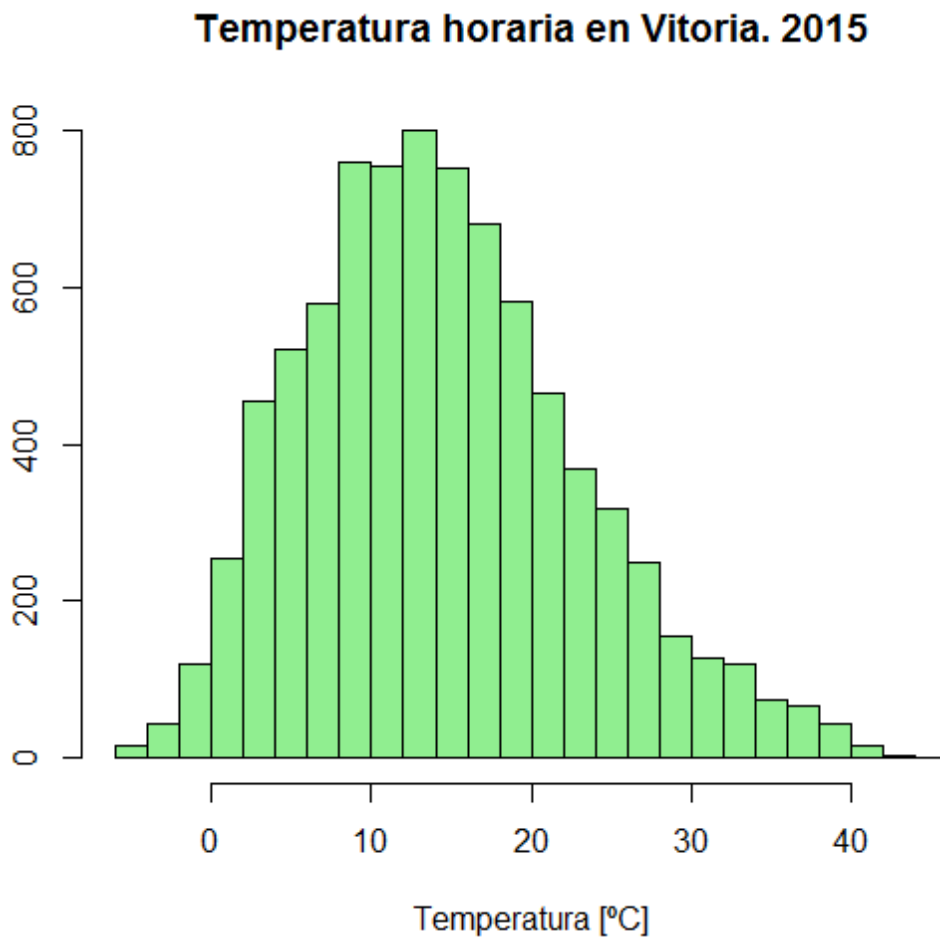
Una vez seleccionados los datos, debemos llevar a cabo un análisis de las propiedades correspondientes. Se tendrán en cuenta histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).

A continuación se muestran varios histogramas y gráficos que nos ayudarán;



Histograma 1: Frecuencia de los valores de la temperatura interior horaria del supermercado en el año 2015.

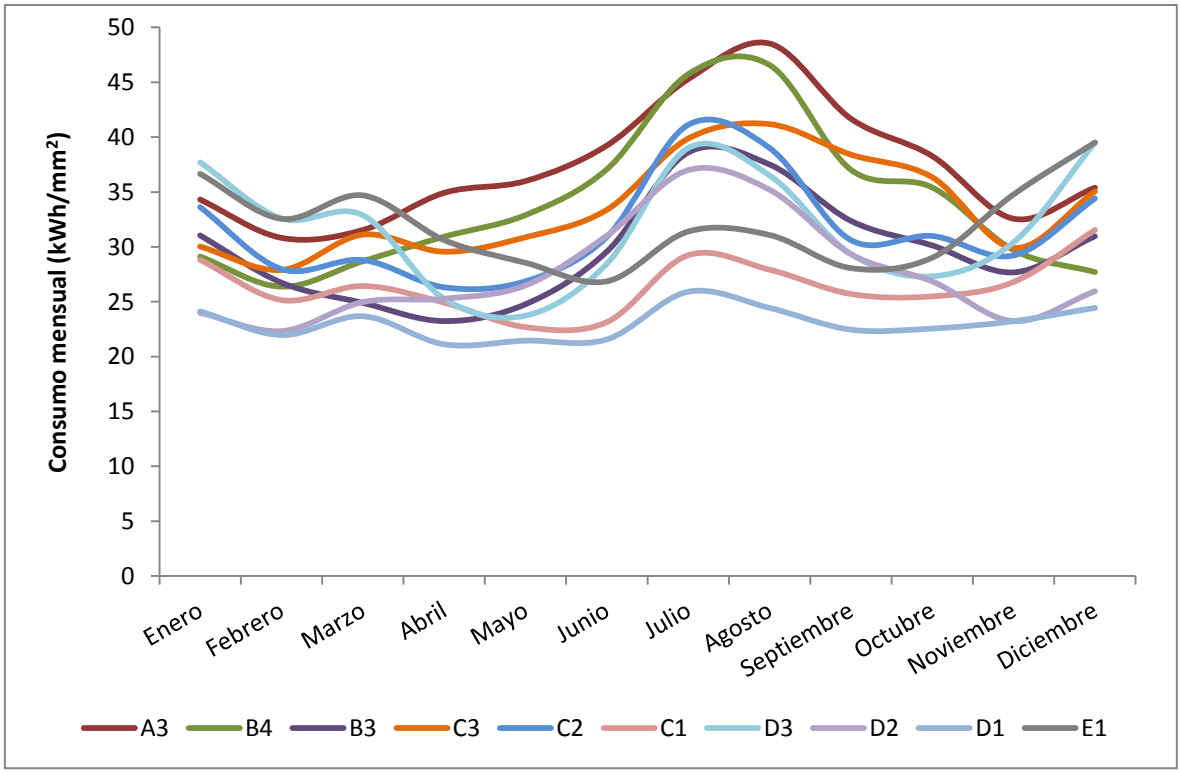
El Histograma 1 muestra los valores de la temperatura (Cº) en el interior del supermercado a lo largo del año 2015. Los valores más comunes son los que van desde los 18 grados hasta los 23 (rango de confort humano). La temperatura tanto en el interior como en el exterior del supermercado afecta directamente al consumo de energía; a mayor temperatura interior, más demanda en calefacción, y contra menor sea la temperatura exterior, más demanda en controlar diferencia de temperatura. A continuación se verá cómo varía el consumo dependiendo de la zona climática donde esté ubicado el supermercado.



Histograma 2: Frecuencia de los valores de la temperatura exterior horaria del supermercado en el año 2015.

En el Histograma 2 podemos ver la frecuencia de valores que adquirió la temperatura en el exterior del supermercado. Cómo cabe esperar, los valores más frecuentes son más bajos que los que nos encontramos en el interior del supermercado. Rondan entre los 10 y 15 grados, lo cual también es comprensible teniendo en cuenta la zona climática de Vitoria, D1. En España podemos distinguir varias zonas climáticas que se identifican mediante una letra, correspondiente a la severidad climática de invierno, y un número, correspondiente a la severidad climática de verano, siendo la zona D1 aquella que se corresponde con inviernos duros y veranos poco calurosos. El consumo eléctrico de los supermercados depende de la zona climática en la que esté situado el establecimiento; en una gráfica que muestre el consumo eléctrico por superficie y por mes a lo largo del año para establecimientos de la misma superficie o parecida veremos que en verano el consumo es mayor en aquellas zonas en las que la temperatura es mayor, ya que aumenta la demanda de frío; en invierno veremos cómo desciende el consumo en esas zonas y aumenta el de las zonas en las que la temperatura en esos meses es baja y se hace necesario calentar el establecimiento.

A modo de ejemplo representativito y para corroborar lo que se acaba de afirmar se ha elaborado la Gráfica 15.

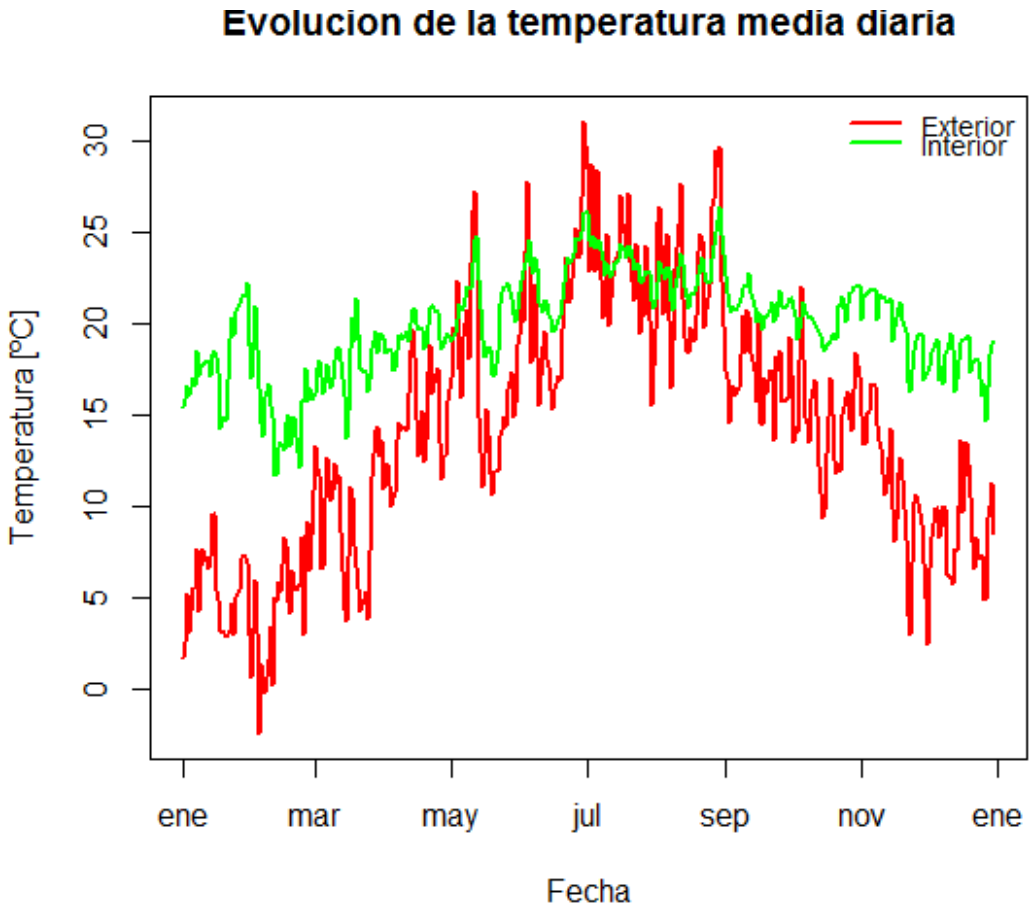


Gráfica 15. Consumo eléctrico por superficie al mes a lo largo del año 2013 en diferentes zonas climáticas de establecimientos de área de superficie de ventas de aproximadamente 5000 m^2

Primero nos vamos a fijar en el consumo promedio mensual en los **meses de verano**. Las zonas en las que los veranos son más duros son lo que tienen el número más alto. En base a esto, apreciamos el mayor consumo para la zona A3 (la más calurosa). Si nos fijamos en los datos de las dos zonas 'B', el consumo es mayor en la B4 que en la B3. Haciendo lo mismo con las zonas 'C', ocurre lo mismo y se cumplen nuestras hipótesis (C3>C2>C1). Para el caso de las zonas 'D' también se cumple que cuando el verano es más caluroso mayor es el consumo.

Ahora vamos a analizar el consumo en los **meses de invierno.** Los inviernos más duros, con temperaturas más bajas se dan en las zonas con la letra 'E', les siguen los de la 'D', 'C', 'B' y las zonas con la denominación 'A'. Efectivamente, el consumo de la zona E1 aumenta en estos meses colocándose entre los valores más altos. Destacamos el aumento que se da en el consumo de la zona D3. En cuanto a las zonas 'C' no vemos grandes variaciones, sobre todo destaca lo estable que se mantiene el consumo en la zona C1. Es una zona en la que las temperaturas son suaves, sin inviernos ni veranos duros. En lo referente a las zonas con los inviernos más suaves 'A' y 'B' se aprecia como el consumo en invierno apenas aumenta, incluso sufre un descenso en el consumo en el supermercado tomado como ejemplo para la zona B4.

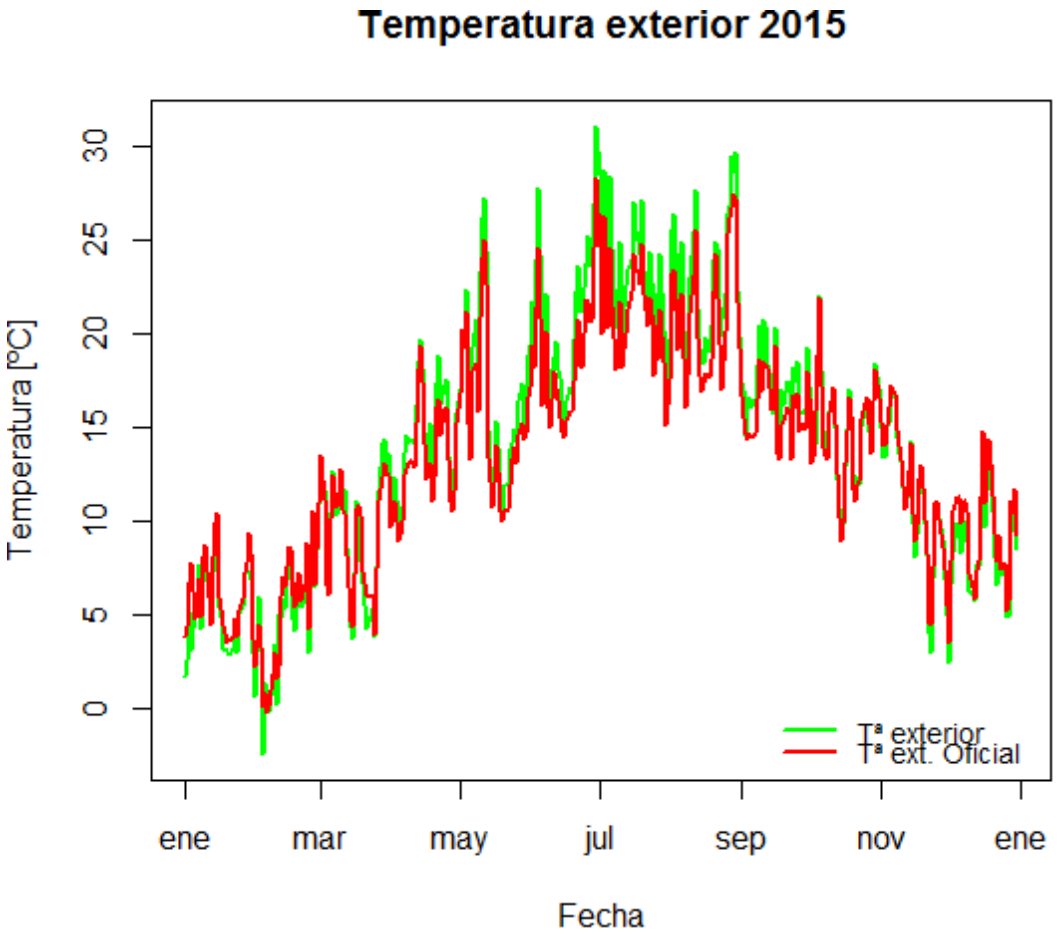
Como vemos en la Gráfica 15 es más acusado el aumento en verano que el de invierno, esto es debido a que el consumo para generar frío es el que mayor peso tiene con respecto al total, como ya hemos explicado anteriormente. También vemos que el consumo eléctrico es mayor, generalmente, en las zonas más cálidas que el de las zonas más frías.



Gráfica 16: Temperatura media diaria interior y temperatura media diaria exterior del año 2015.

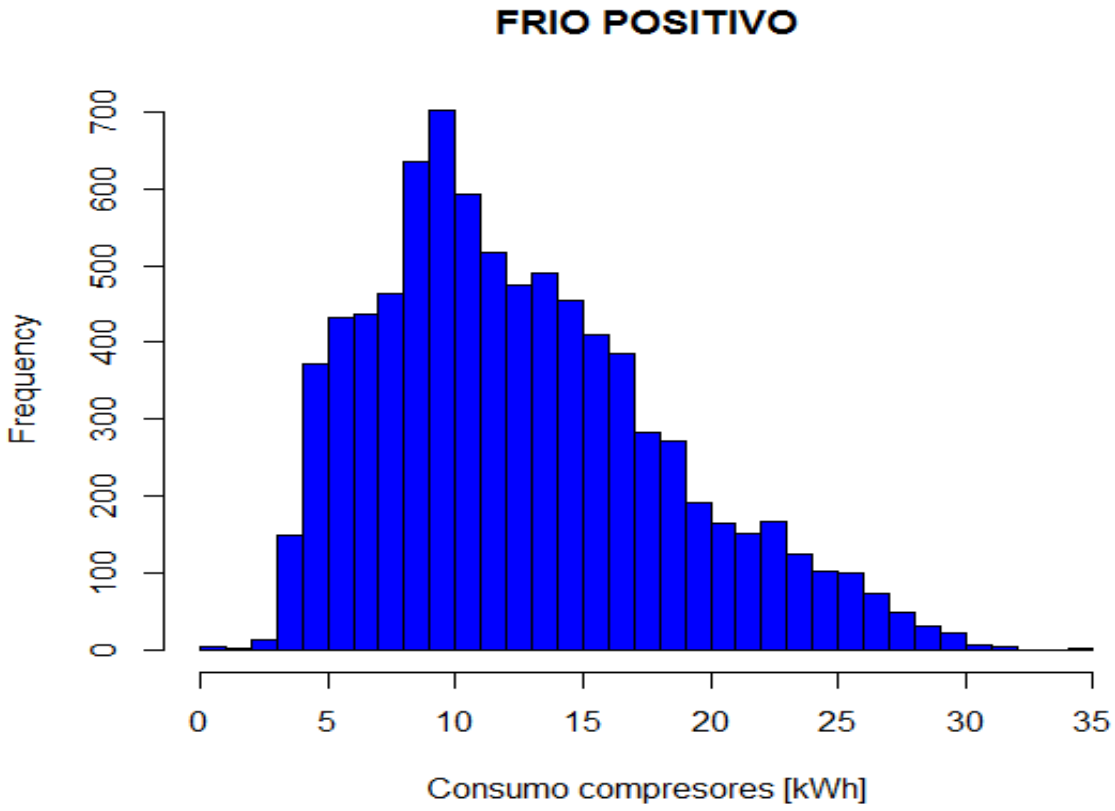
En la Gráfica 16 se muestran tanto la temperatura en el interior cómo en el exterior a lo largo del año de manera que se puedan comparar fácilmente entre sí.

La temperatura en el interior del supermercado es más constante que en el exterior, por lo que a la hora de predecir los datos será importante tener en cuenta ambas, ya que sólo con la interior no tenemos suficiente información. A priori de hecho dicha variable no parece que pueda influir demasiado en el resultado, pero hay que tener en cuenta que combinando distintas variables se pueden obtener relaciones e información adicional que no se observa a simple vista.

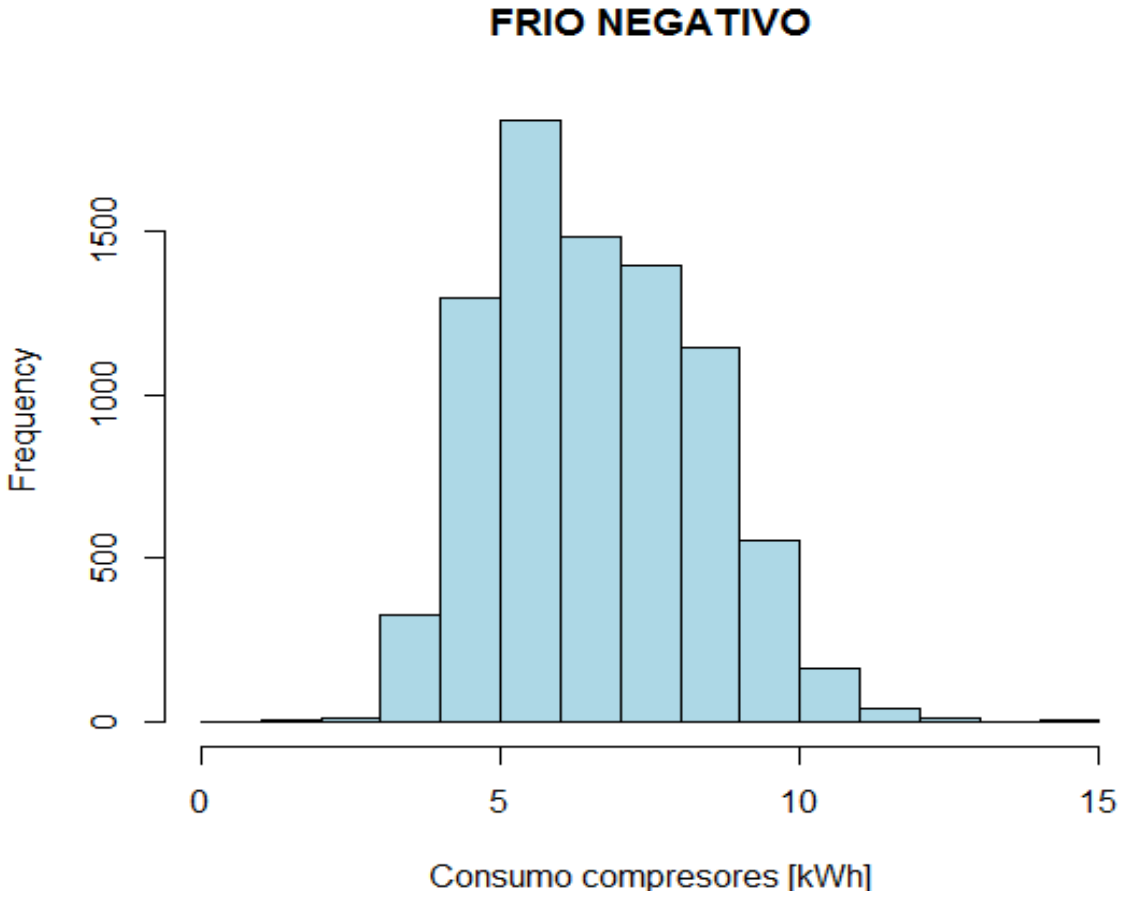


Gráfica 17: Temperatura media diaria en el exterior procedente de los analizadores de CENER y temperatura media diaria en el exterior procedente de los datos de EUSKALMET.

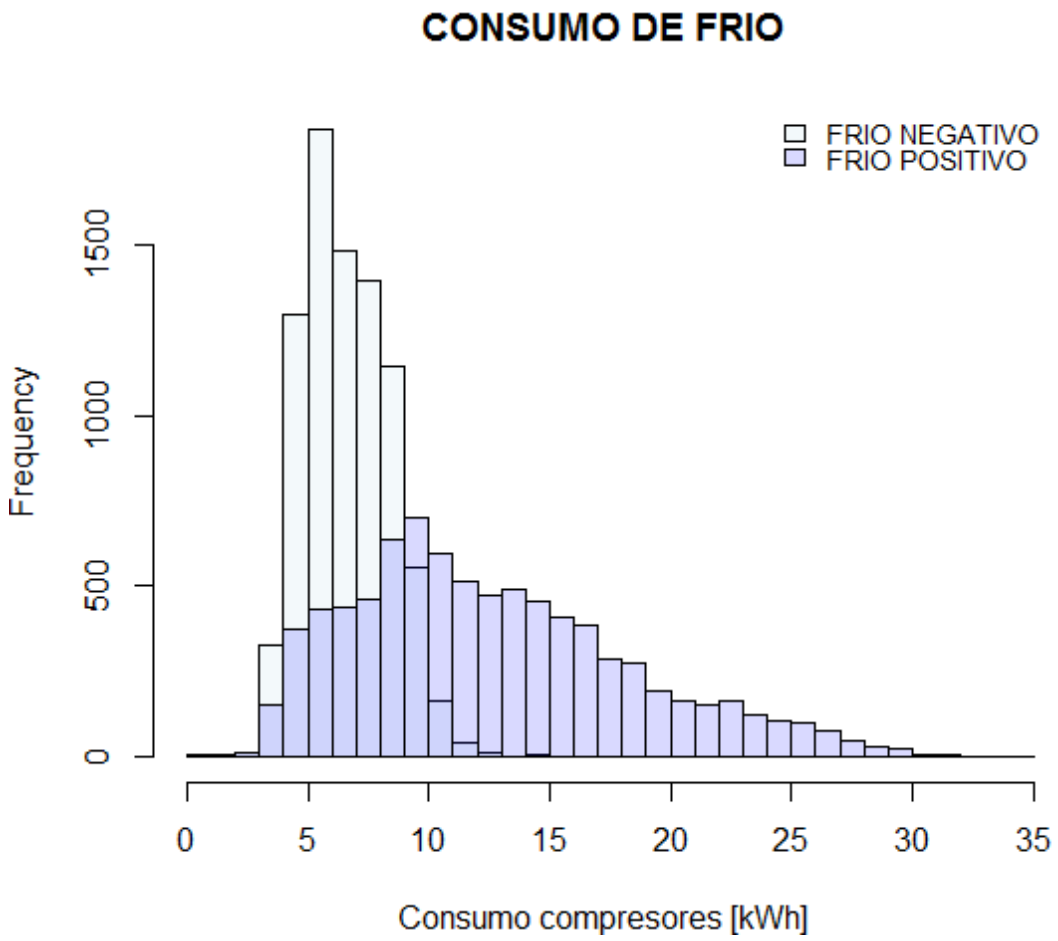
La Gráfica 17 nos muestra la temperatura en el exterior a lo largo del año obtenida mediante los dispositivos colocados por el centro junto con los datos de la misma proporcionados por EUSKALMET. Esto nos sirve para comprobar si la temperatura tiene valores correctos y las mediciones se estaban realizando correctamente. Como se puede observar, parece que las mediciones son correctas ya que ambas representaciones son muy similares. Podemos asegurar por tanto que los analizadores de temperatura funcionan correctamente en la posición en que fueron colocados y que las medidas que proporcionan son suficientemente coherentes y precisas cómo para utilizar dichos valores.



Histograma 3: Consumo de los compresores de frío positivo (kWh).



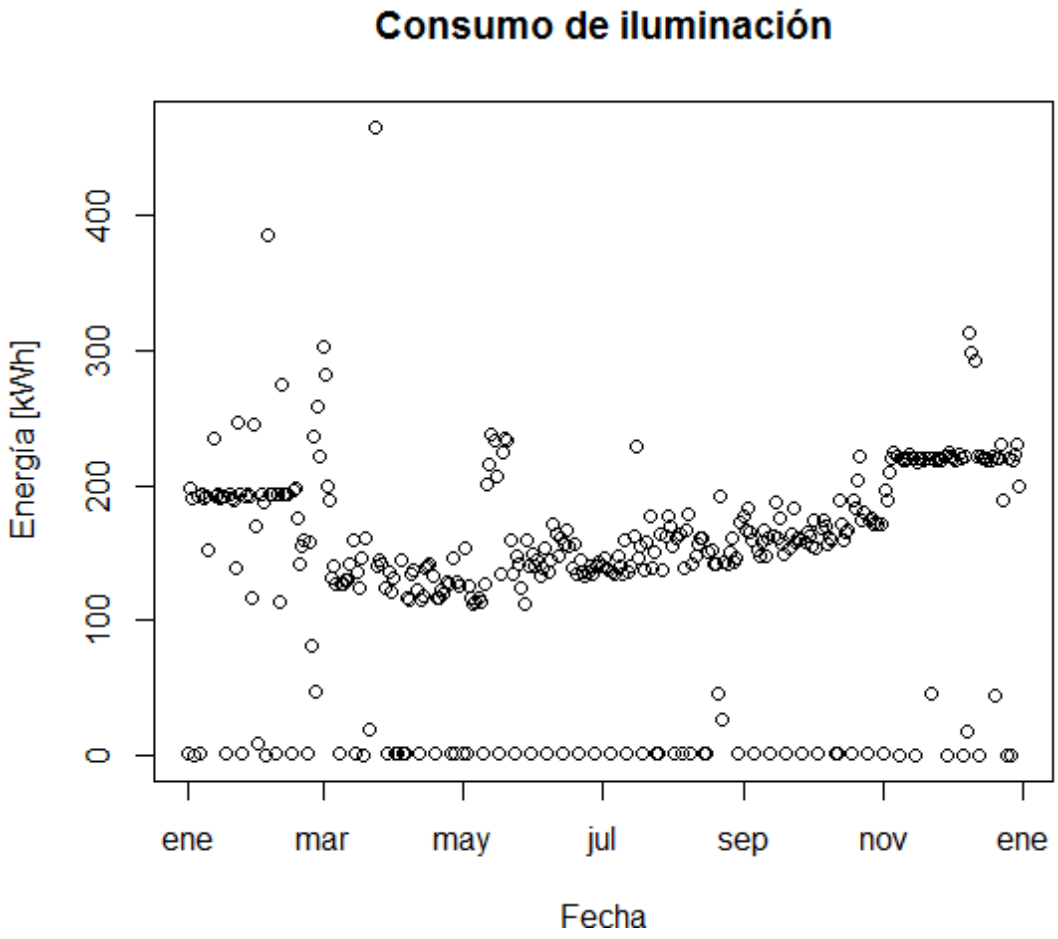
Histograma 4: Consumo de los compresores de frío negativo (kWh).



Histograma 5: Consumo de los compresores de frío positivo y negativo (kWh).

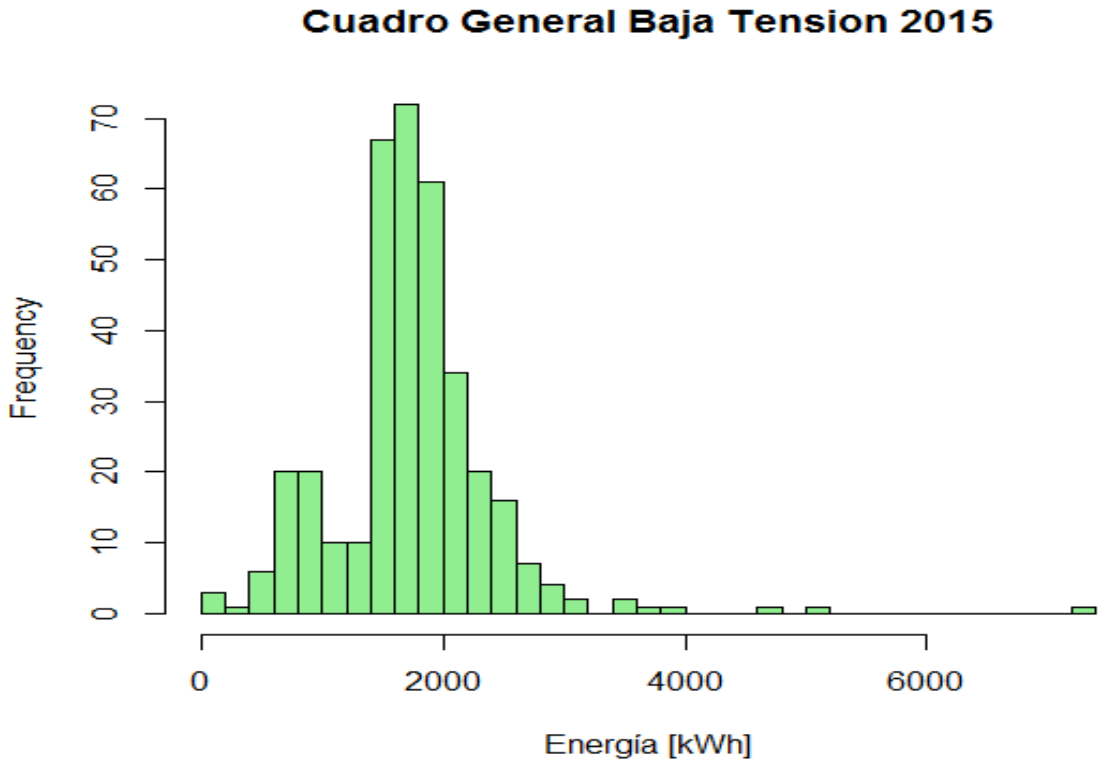
En los Histogramas 3 y 4 se muestran los valores más frecuentes del consumo por frío positivo y negativo respectivamente. La conservación por frío positivo es aquella que va por encima de los 0°C (de 1 a 3°C o entre 4 y 6°C dependiendo de los alimentos) y se conoce comúnmente como refrigeración. La conservación por frío negativo es aquella que va por debajo de los 0°C y es la que conocemos como congelación.

En el Histograma 5 se muestran ambos a la vez, y se comprueba que el consumo de frio negativo es más constante y que el de frio positivo varía mucho a lo largo del año. El consumo energético por frío es una variable que a priori influirá de gran manera en los modelos que se generen, ya que la variable a predecir mide el consumo energético, y una gran cantidad de consumo proviene precisamente de la refrigeración. Es por esto también por lo que se deberá estudiar la correlación entre variables y la importancia de cada una y así poder comparar la influencia real de cada variable de temperatura, consumo, etc.



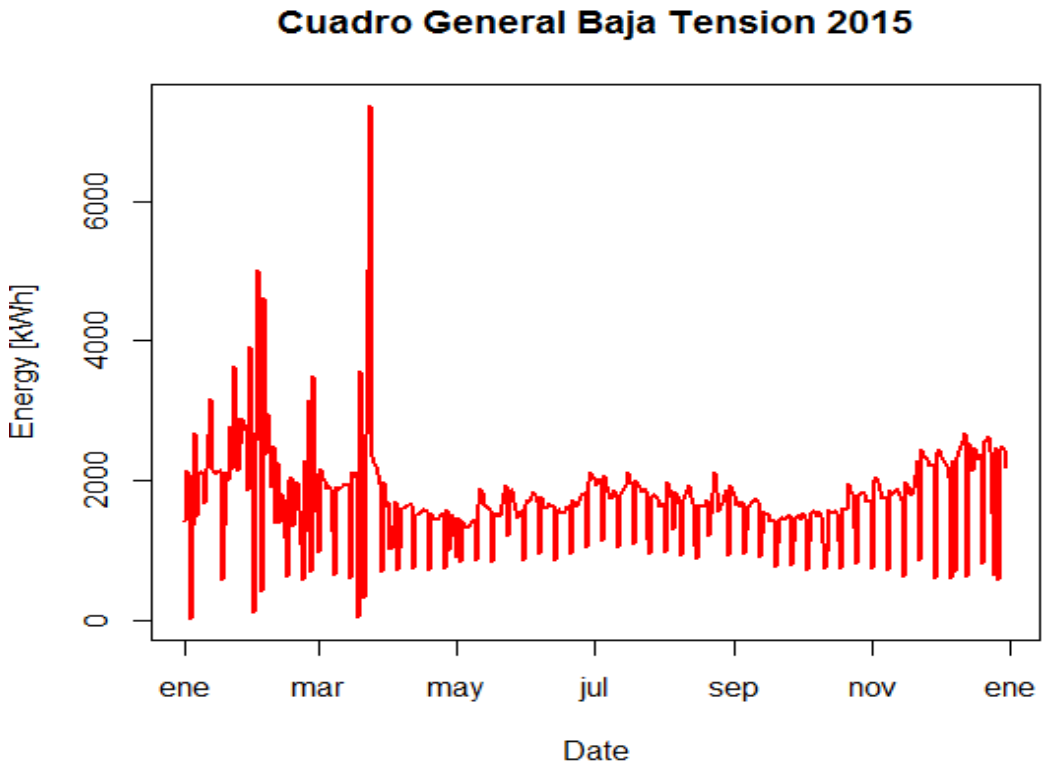
Gráfica 18: Consumo medio diario por iluminación (kWh).

Otro parámetro muy importante a la hora de estimar consumo eléctrico es el consumo por iluminación. En la Gráfica 18 podemos ver el consumo en kWh a lo largo del año. Los valores para los que no hay consumo no son errores, sino que se trata de los días festivos en los que el supermercado no estaba abierto y por tanto no hubo consumo de iluminación. Esto nos servirá para la creación de la variable *laboralidad* comentada con anterioridad y para la clasificación entre días laborales y festivos. Al igual que para las variables de frío positivo y negativo, la iluminación repercute directamente en la cantidad total de energía consumida por el supermercado y por tanto dicho valor variará conforme varíen los valores de iluminación.



Histograma 6: Consumo del Cuadro General de Baja Tensión (kWh) del supermercado en 2015.

El Histograma 6 muestra los valores más frecuentes de la variable que nos interesa estudiar y predecir. Podemos observar claramente que los datos no están muy centralizados. Debemos analizar si los datos más extremos son valores correctos o si por el contrario algo falló a la hora de medir y son por tanto son valores anómalos que se deben eliminar.



Gráfica 19: Consumo diario del Cuadro General de Baja Tensión en el supermercado en 2015.

En la Gráfica 19 se pueden observar una serie de datos que llaman la atención debido a que se distinguen más fácilmente que el resto. Existe una cierta regularidad a lo largo de los datos pero también hay ejemplos que se distancian mucho de la media. Estos valores podrían ser valores anómalos o "outliers", es decir, instancias que ya sea por algún error en la medición o en algún cálculo, o por alguna razón no habitual que se dio en un momento puntual, son erróneas y se deben eliminar ya que en caso contrario podrían dificultar el aprendizaje y empeorar los resultados. Como hemos visto en las figuras anteriores, para el caso de la temperatura no encontrábamos valores dispares en el primer cuatrimestre del año, y tampoco en el consumo de frío o iluminación, por lo que en primera instancia podemos pensar que ha sucedido algo fuera de lo común y los valores pueden no ser correctos. Es por esto que a continuación se realizan una serie de pasos que nos aportarán una mayor información y nos ayudará a la hora de decidir si dichos valores deben ser eliminados o no.

En la Figura 5 podemos ver estos valores distanciados mediante un diagrama de caja (boxplot).

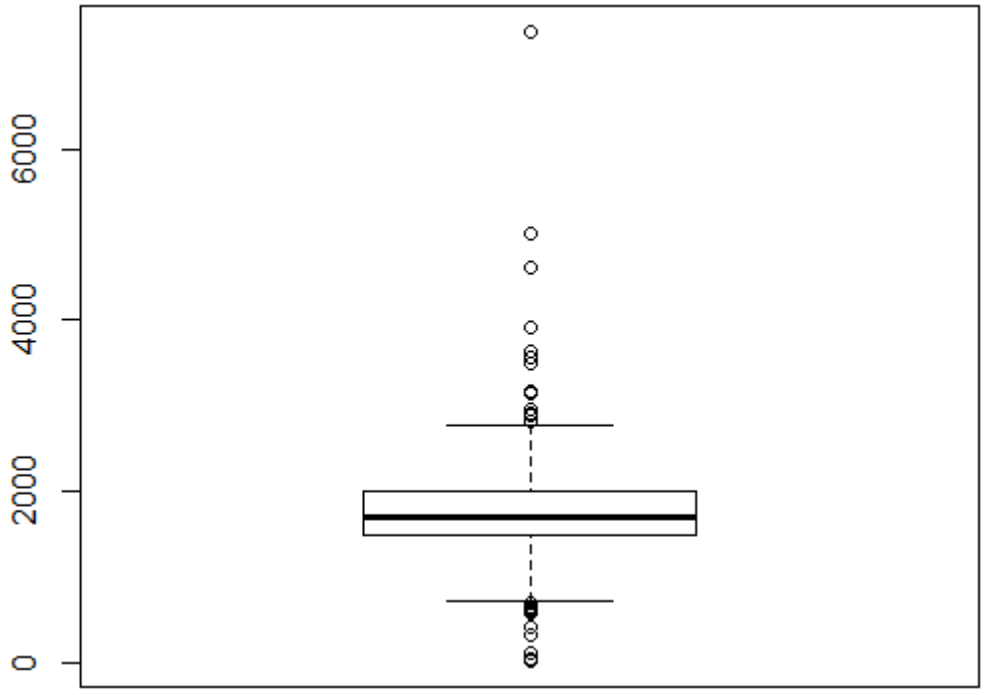


Figura 5: Diagrama de caja de los valores asociados al consumo del Cuadro General de Baja Tensión del supermercado en 2015.

Este es un gráfico basado en cuartiles y mediante el cual se visualiza la distribución de los datos y es útil para ver la presencia de valores atípicos. En un diagrama de caja como este, tomando como referencia la diferencia entre el primer cuartil y el tercer cuartil, o valor intercuartil, se considera un valor atípico el que se encuentra 1,5 veces esa distancia de uno de esos cuartiles (atípico leve) o a 3 veces esa distancia (atípico extremo).

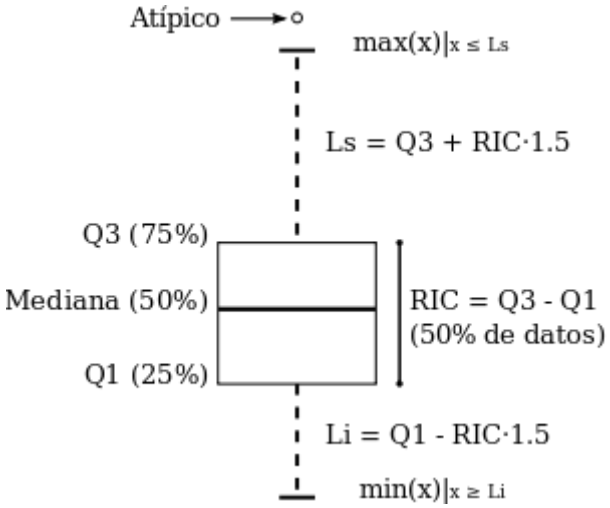


Figura 6: Estructura de un Diagrama de Caja o *Boxplot.*

A la vista de los resultados, hemos decidido que hay ciertos datos que deben ser eliminados. No obstante, dichos datos se utilizarán para la creación de un primer modelo de manera que sirva de base para posteriores comparaciones con modelos en los que los valores atípicos se han eliminado. De esta forma podemos desechar ciertas configuraciones en caso de que no superen al modelo base.

El siguiente paso es la transformación del conjunto de datos de entrada de manera que se pueda aplicar correcta y eficazmente la o las distintas técnicas de minería de datos, ya que, para poder aplicar correctamente algunas técnicas como puede ser la técnica de red neuronal, es necesario re-escalar o normalizar los datos. Dicho esto, la normalización no es siempre estrictamente necesaria, ya que depende de la variabilidad de los datos y como se distribuyen. Para este caso es vital ya que las variables que vamos a utilizar para predecir el consumo tienen una escala muy distinta. Podremos utilizar tres tipos distintos de normalización:

- Normalización de max-min. { (x - min(x))/(max(x) - min(x)) } (ejecuta una transformación lineal de los datos originales, con base en los valores mínimos y máximos de un atributo, se calcula un valor de normalización v' con base en el valor v. Este método conserva las relaciones entre los datos originales. Los valores estarán en el rango [0,1]).
- Normalización similar a la de max-min pero que se utiliza cuando se desean unos datos más centralizados; { x-((max+min)/2) / (max-min)/2 }
- Normalización estándar o z-score: los valores para un atributo A son normalizados basados en la media y la desviación estándar de A. Este método se utiliza cuando el máximo y el mínimo son desconocidos o cuando hay valores anómalos que predominan al usar la normalización min-max.

Otra cosa importante es la correlación entre variables. El coeficiente de correlación entre dos variables es la covarianza dividida entre el producto de sus desviaciones estándar. Nos indica lo linealmente dependientes que son unas variables de otras. Si se acerca a 1, tienen una

relación lineal fuerte. Si se acerca a 0, es una relación débil y si se acerca a -1 no están relacionadas.

En nuestro caso es interesante conocer la correlación entre las variables de manera que si detectamos alguna dependencia lineal muy fuerte se elimine dicha variable o variables ya que no nos aportan nada nuevo.

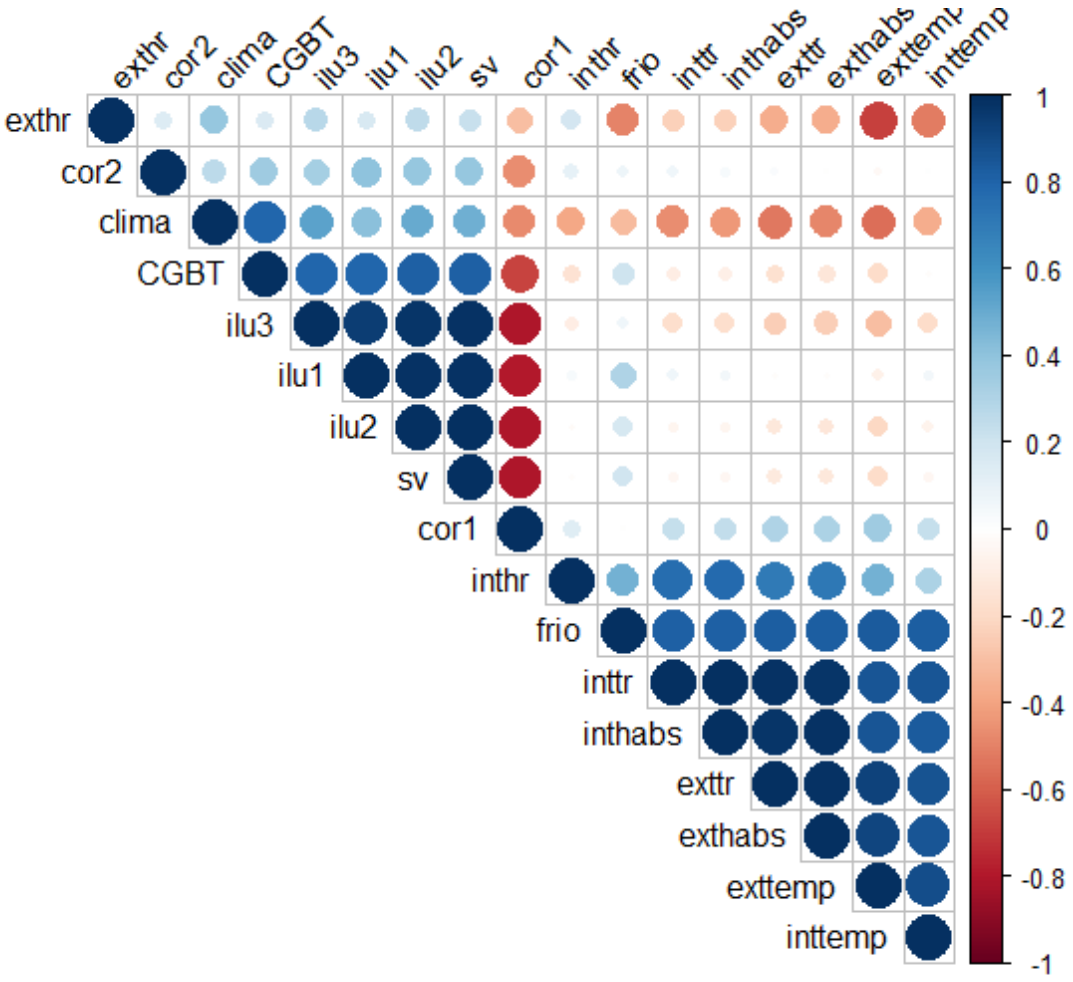


Figura 7: Correlación entre las variables objeto de estudio del modelo representada de forma gráfica.

Con la información de la correlación podemos observar que la variable CGBT que nos interesa tiene una relación lineal muy fuerte con las variables ilu1, ilu2, ilu3 y sv, las cuales representan el consumo energético proveniente del alumbrado de diferentes secciones del supermercado.

Para llevar a cabo una selección de variables no se dispone únicamente de la correlación. El algoritmo más utilizado es el PCA (Análisis de componentes principales). Es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos lo máximo posible perdiendo la menor cantidad de información posible. Los nuevos componentes principales que se consiguen crear son combinaciones lineales de los datos originales y son independientes entre sí. Aplicar el PCA tiene sentido si existen altas correlaciones entre las variables (existe información redundante) y por tanto con menos variables se puede explicar la mayor parte de la variabilidad. Los nuevos factores se crean de manera que el primero recoja la mayor

proporción posible de la variabilidad original, el segundo la mayor variabilidad no recogida por el primero y así sucesivamente. Los nuevos factores principales serán aquellos con los que se recoja la proporción suficiente (suele ser el 95% de la variabilidad original).

Una vez llevado a cabo el proceso de normalización de los datos de entrada, la limpieza de valores erróneos y /o atípicos y la selección de variables a utilizar, el siguiente paso es definir el modelo al que se aplicarán las distintas técnicas. Uno de los pasos más importantes es la creación del conjunto de datos de entrenamiento y el conjunto de datos de prueba.

Existen diversas técnicas y opciones, ya que sobretodo esta división se debe llevar a cabo teniendo en cuenta los datos a los que se aplica. Una de las técnicas más utilizadas es la de la Validación Cruzada.

La validación cruzada o cross-validation es una técnica utilizada para validar modelos generados. Evalúa los resultados de un análisis estadístico y garantiza que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir la medida de evaluación con distintas particiones y calcular la media aritmética. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar cómo de preciso es un modelo real, práctico. La validación cruzada proviene de la mejora del método de retención o *holdout method* (Ilustración 7). Este consiste en dividir en dos conjuntos complementarios los datos de muestra, realizar el análisis de un subconjunto (denominado datos de entrenamiento o *training set*), y validar el análisis en el otro subconjunto (denominado datos de prueba o *test set*), de forma que la función de aproximación sólo se ajusta con el conjunto de datos de entrenamiento y a partir de aquí calcula los valores de salida para el conjunto de datos de prueba. La ventaja de este método es que es muy rápido a la hora de computar. Sin embargo, este método no es demasiado preciso debido a la variación del resultado obtenido para diferentes datos de entrenamiento. La evaluación puede depender en gran medida de cómo es la división entre datos de entrenamiento y de prueba, y por lo tanto puede ser significativamente diferente en función de cómo se realice esta división. Debido a estas carencias aparece el concepto de validación cruzada

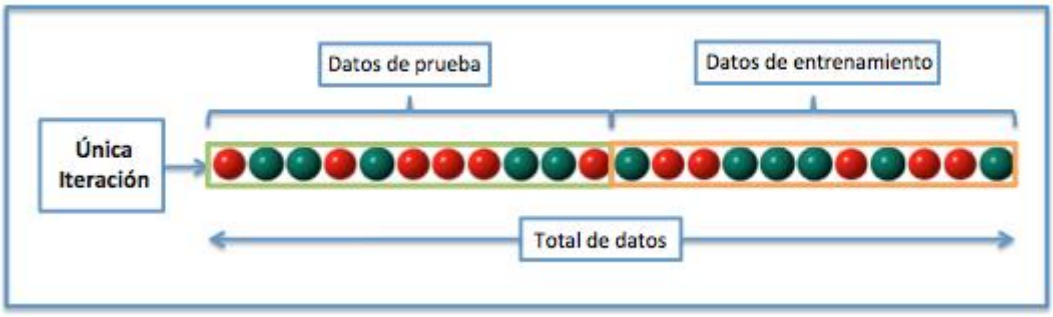


Ilustración 7: Método de retención o *Holdout.*

Objetivo de la validación cruzada

Suponemos que tenemos un modelo con uno o más parámetros de ajuste desconocidos y unos datos de entrenamiento que queremos analizar. El proceso de ajuste optimiza los parámetros del modelo para que éste se ajuste a los datos de entrenamiento tan bien como pueda. Si cogemos una muestra independiente como dato de prueba (validación), del mismo grupo que

los datos de entrenamiento, normalmente el modelo no se ajustará a los datos de prueba igual de bien que a los datos de entrenamiento. Esto se denomina sobreajuste y acostumbra a pasar cuando el tamaño de los datos de entrenamiento es pequeño o cuando el número de parámetros del modelo es grande. La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba.

Existen varios tipos o formas de llevar a cabo la validación cruzada. A continuación podemos ver algunas;

Validación cruzada de K iteraciones

En la validación cruzada de K iteraciones o *K-fold cross-validation* los datos de muestra se dividen en K particiones (Ilustración 8). Una de las particiones se utiliza como *test* y las K-1 restantes como *train*. Esto se repite K veces de manera que cada una de las particiones se haya utilizado como *test* y las restantes como *train*. Por último se realiza la media aritmética de todos los resultados de cada iteración y se obtiene un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos.

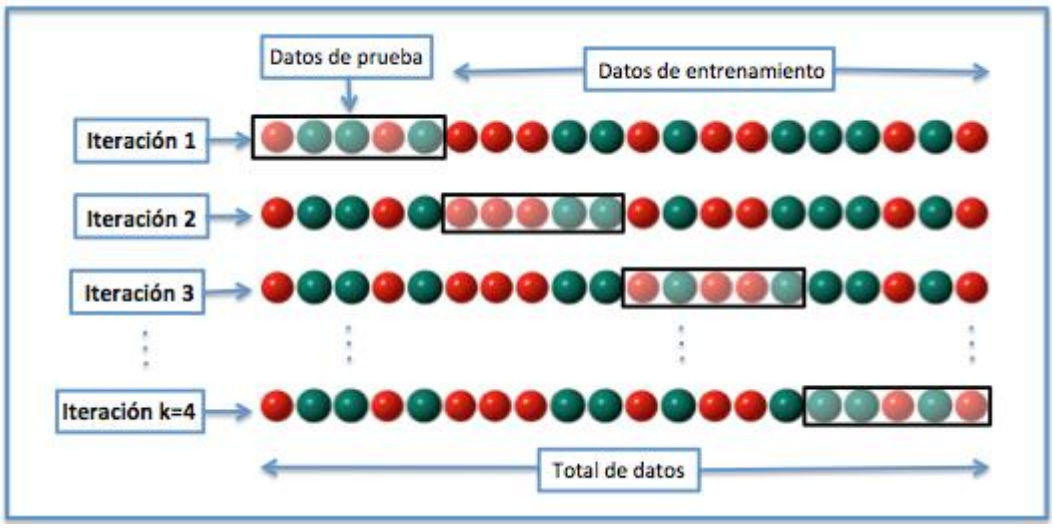


Ilustración 8: Validación cruzada de K = 4 iteraciones.

Validación cruzada aleatoria

Este método consiste en dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba (Ilustración 9). Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones. La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.

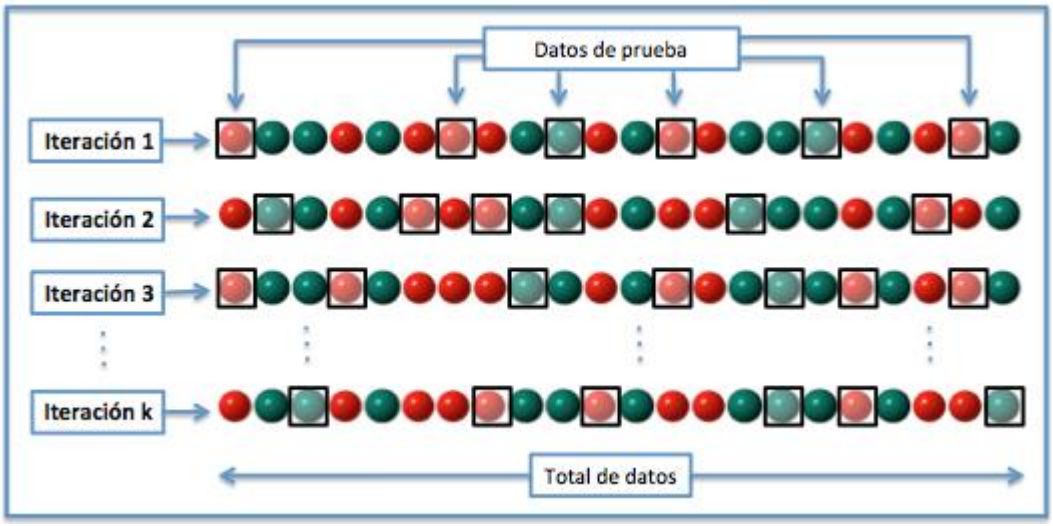


Ilustración 9: Validación cruzada aleatoria con K iteraciones.

Validación cruzada dejando uno fuera

La validación cruzada dejando uno fuera o *Leave-one-out cross-validation (LOOCV)* implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento (Ilustración 10). La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

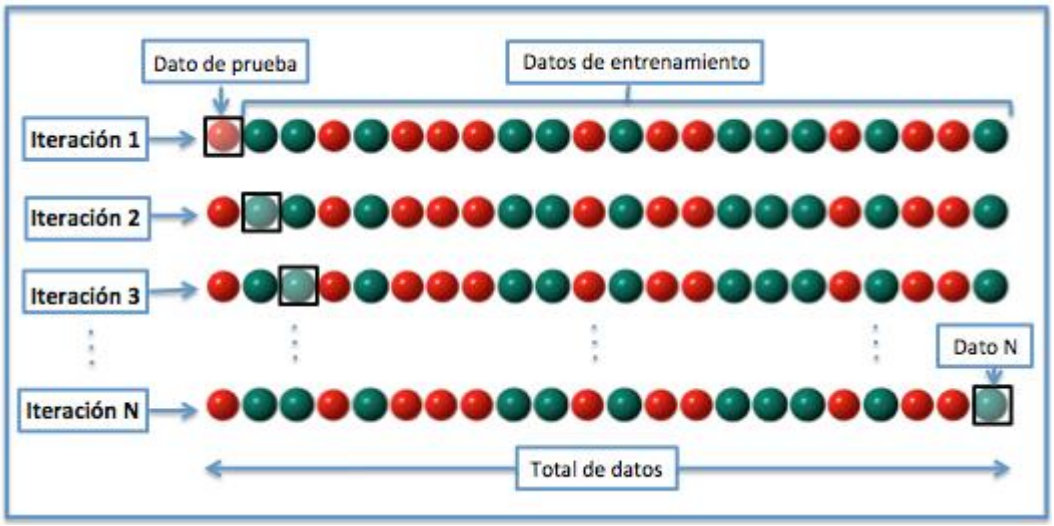


Ilustración 10: Validación cruzada dejando uno fuera (LOOCV)

Ejemplos de aplicación

La validación cruzada se puede utilizar para comparar los resultados de diferentes procedimientos de clasificación predictiva. Por ejemplo, supongamos que tenemos un detector que nos determina si una cara pertenece a una mujer o a un hombre y consideramos que han sido utilizados dos métodos diferentes, por ejemplo, máquinas de vectores de soporte (SVM) y

K-vecinos más cercanos (Knn), ya que ambos nos permiten clasificar las imágenes. Con la validación cruzada podríamos comparar los dos procedimientos y determinar cuál de los dos es el más preciso. Esta información nos la proporciona la tasa de error que obtenemos al aplicar la validación cruzada por cada uno de los métodos planteados.

La validación cruzada de "k" iteraciones (*k-fold cross validation*) nos permite evaluar también modelos en los que se utilizan varios clasificadores. Continuando con el ejemplo anterior, si tenemos un detector que nos determina si en una imagen aparece un hombre o una mujer, y éste utiliza cuatro clasificadores binarios para detectarlo, también podemos utilizar la validación cruzada para evaluar su precisión. Si tenemos un total de 20 datos (imágenes), y utilizamos el método *4-fold cross validation*, se llevarán a cabo cuatro iteraciones, y en cada una se utilizarán unos datos de entrenamiento diferentes, que serán analizadas por cuatro clasificadores, que posteriormente evaluarán los datos de prueba. De este modo por cada muestra obtendremos cuatro resultados, y si hacemos la media entre los resultados de cada clasificador y entre las cuatro iteraciones realizadas, obtendremos el valor resultante final.

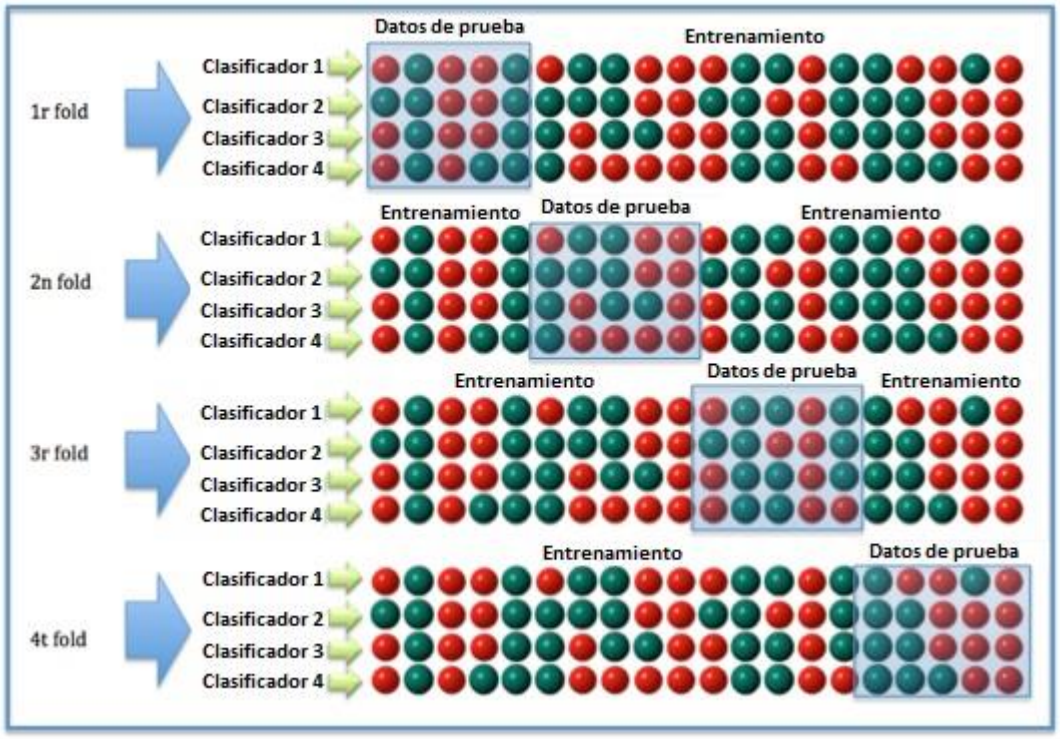


Ilustración 11: K-fold cross validation, con K = 4 y 4 clasificadores.

Hay que tener en cuenta que la validación cruzada sólo produce resultados significativos si el conjunto de validación y prueba de conjunto se han extraído de la misma población. En muchas aplicaciones de modelado predictivo, la estructura del sistema que está siendo estudiado evoluciona con el tiempo. Esto puede introducir diferencias sistemáticas entre los conjuntos de entrenamiento y validación. Por ejemplo, si un modelo para predecir el valor de las acciones está entrenado con los datos de un período de cinco años determinado, no es realista para tratar el siguiente período de cinco años como predictor de la misma población.

Otro ejemplo, supongamos que se desarrolla un modelo para predecir el riesgo de un individuo para ser diagnosticado con una enfermedad en particular en el próximo año. Si el

modelo se entrena con datos de un estudio que sólo afecten a un grupo poblacional específico (por ejemplo, solo jóvenes o solo hombres varones), pero se aplica luego a la población en general, los resultados de la validación cruzada del conjunto de entrenamiento podrían diferir en gran medida de la clasificación real.

Si se lleva a cabo correctamente, y si el conjunto de validación y de conjunto de entrenamiento son de la misma población, la validación cruzada es casi imparcial. Sin embargo, hay muchas maneras en que la validación cruzada puede ser mal utilizada. Si se abusa y posteriormente se lleva a cabo un estudio real de validación, es probable que los errores de predicción en la validación real sean mucho peores de lo esperado sobre la base de los resultados de la validación cruzada.

Estas son algunas formas en que la validación cruzada puede ser mal utilizada:

- Mediante el uso de la validación cruzada para evaluar varios modelos, y sólo indicando los resultados para el modelo con los mejores resultados.

- Al realizar un análisis inicial para identificar las características más informativas utilizando el conjunto de datos completo, si la selección de característica o el ajuste del modelo lo requiere por el propio procedimiento de modelado, esto debe repetirse en cada conjunto de entrenamiento. Si se utiliza la validación cruzada para decidir qué características se van a utilizar, se deberá realizar un proceso interno de validación cruzada para llevar a cabo la selección de características en cada conjunto de entrenamiento.

- Al permitir que algunos de los datos de entrenamiento estén también incluidos en el conjunto de prueba con lo que varias muestras exactamente idénticas o casi idénticas pueden estar presentes en el conjunto de datos.

Ahora ya podemos dar paso a la tarea realizada; en primer lugar se explicará las distintas divisiones del conjunto de datos. En el proyecto se utilizarán dos conjuntos de datos para la creación de los distintos modelos y estos dos conjuntos tendrán a su vez cuatro posibles clasificaciones o subconjuntos. Los conjuntos de datos son los siguientes:

- Conjunto de datos originales.
- Conjunto de datos filtrados, sin las instancias erróneas o *outliers.*

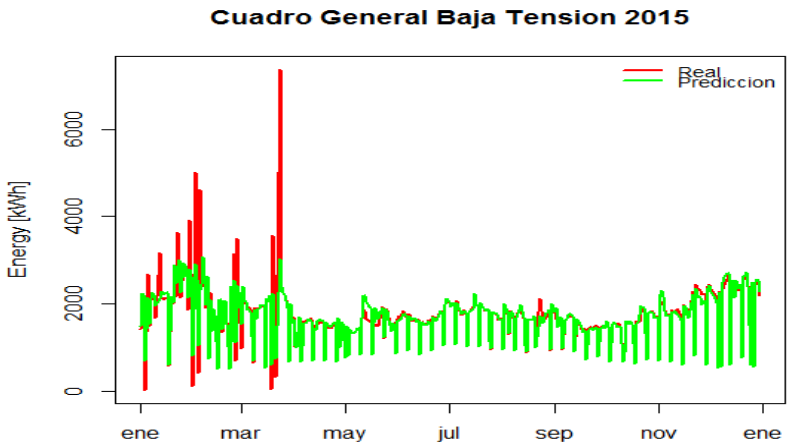
Las versiones para cada conjunto son:

- Crear dos subconjuntos, *train* y *test,* para el entrenamiento y validación del modelo.
- Dividir los datos en dos grupos; fines de semana y restantes. (con el objetivo de observar si este hecho del número de ventas hace que aumente o reduzca el consumo)
- Dividir los datos en dos grupos; días laborales y festivos (con el objetivo de observar cómo afecta el hecho de que el supermercado esté abierto o cerrado).
- Crear dos subconjuntos, *train* y *test,* pero añadir un atributo o variable extra denominada *laboralidad* que se encuentra en el intervalo (0, 1] y mide el grado de "trabajo".

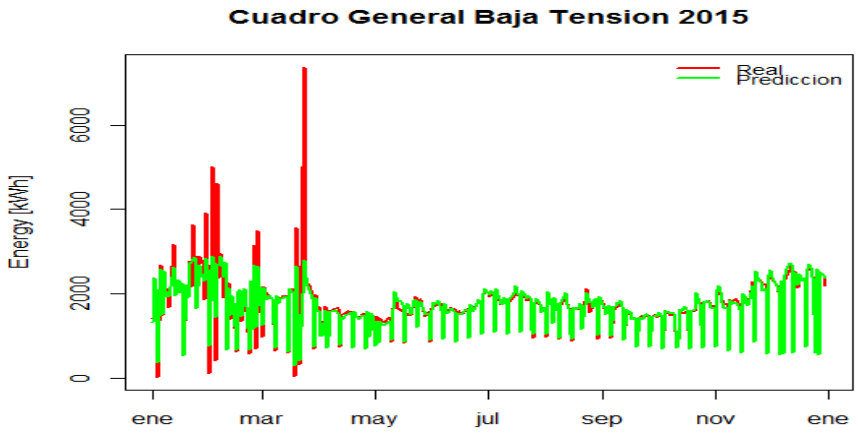
Para empezar, se crea un primer modelo en el que se utilizan todos los datos, es decir, valores atípicos incluidos, que nos servirá de modelo base y punto de partida. Los datos se normalizan con una de las tres técnicas disponibles (normalización de máximo y mínimo, máximo y mínimo centralizados o z-score) de manera que tengamos todos los valores de cada variable y de cada instancia entre 0 y 1 o entre -1 y 1, es decir, todos dentro de un mismo rango de valores, y así poder compararlos fácilmente. A continuación debemos construir el modelo de predicción; podemos utilizar técnicas de selección de variables como PCA para reducir el número que se utilizan, crear distintos conjuntos de datos para entrenar y testear con validación cruzada, etc.

Con la primera prueba que consiste únicamente en seleccionar de forma aleatoria de todos los datos (conjunto original) un 70% de manera que estos formen el conjunto de entrenamiento y el 30% restante para el conjunto de prueba se creará el primer modelo, y nos servirá de referencia de forma que cada uno de los siguientes refleje las mejoras claramente.

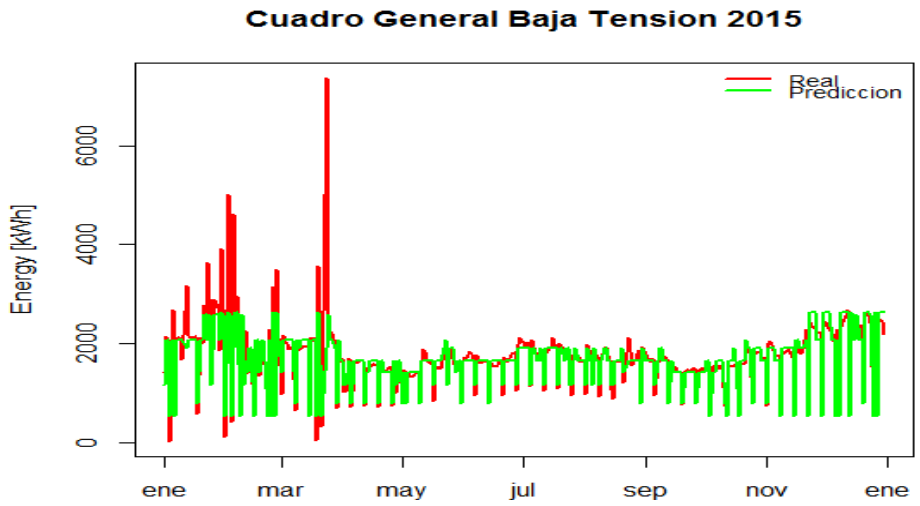
Así pues, ahora se aplican la o las distintas técnicas de minería de datos que correspondan para obtener el modelo de conocimiento. La primera técnica utilizada es la regresión lineal. No se aplica ninguna otra técnica o configuración como cross-validation o PCA. En la Gráfica 20 podemos ver los datos predichos frente a los datos reales que se obtiene.



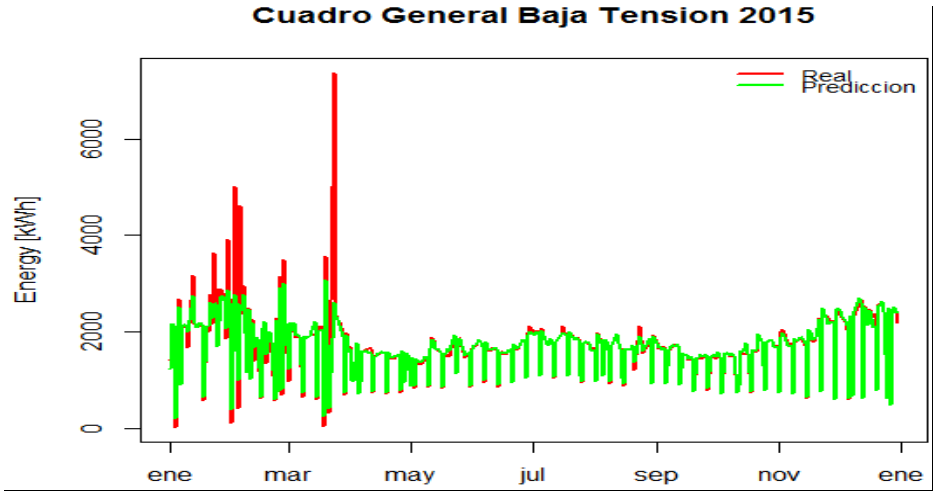
Gráfica 20: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos originales.



Gráfica 21: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos originales.

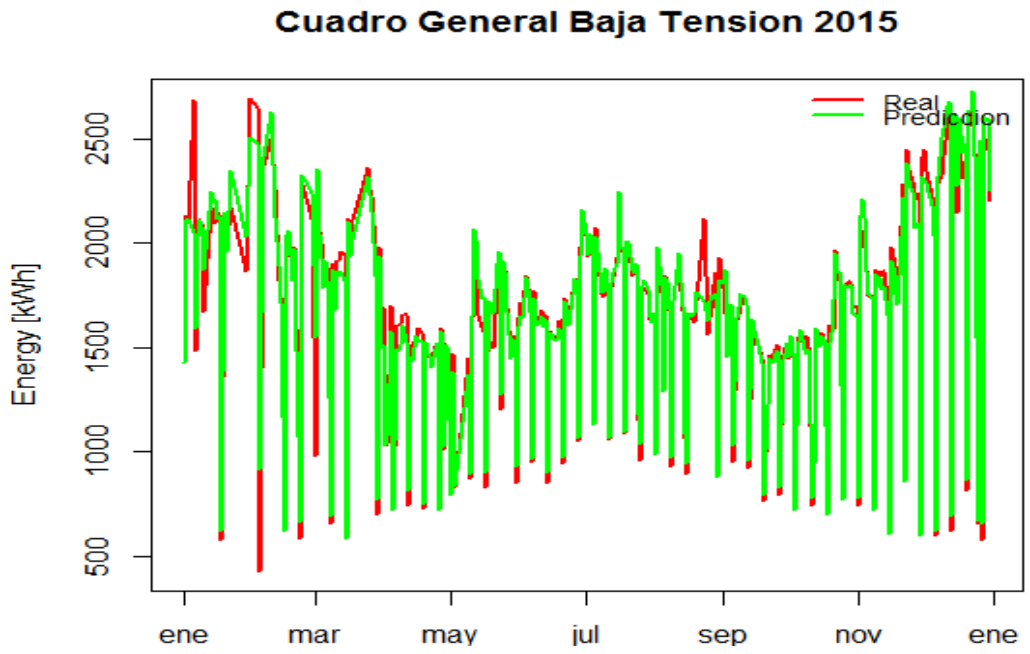


Gráfica 22: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos originales.

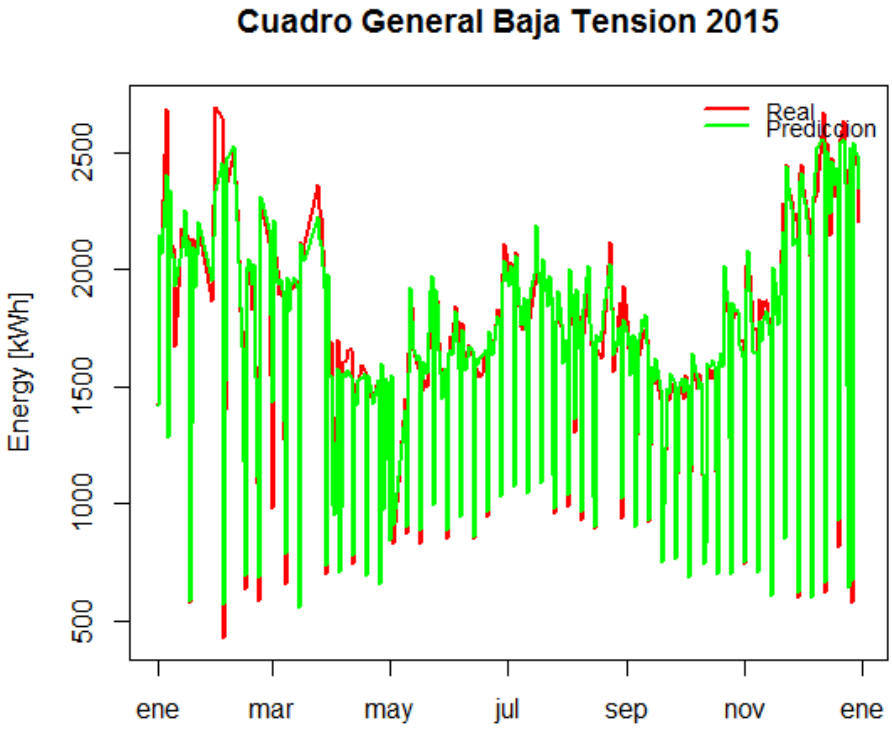


Gráfica 23: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos originales.

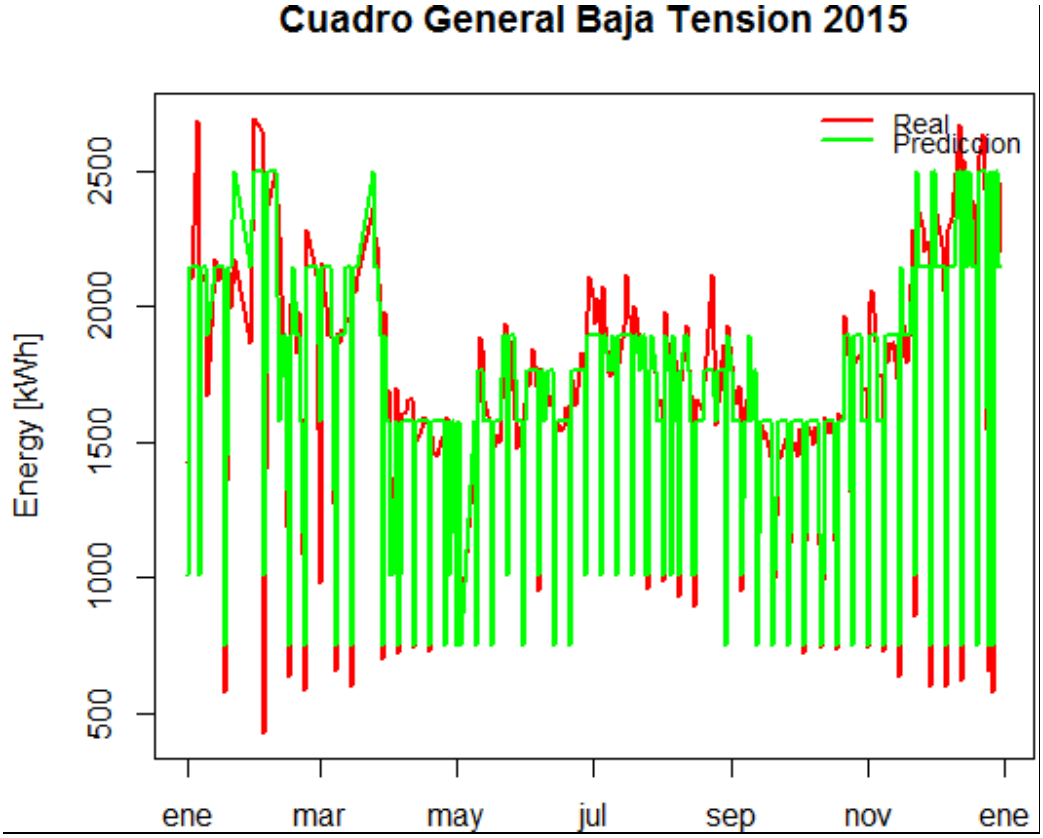
Estos son los cuatro modelos realizados para los datos originales y que nos servirán de punto de partida y referencia. A partir de este punto, se utilizarán los datos ya filtrados, ya que no tiene mucho sentido seguir aplicando mejoras y cambios de configuraciones y parámetros a unos datos que sabemos tienen errores. Las gráficas del resto de pruebas para este conjunto de datos las podremos ver en el anexo pero como se ha comentado, no se tendrán en cuenta. Cómo se ha comentado, utilizamos las técnicas vistas y distintas versiones pero para el siguiente conjunto de datos;



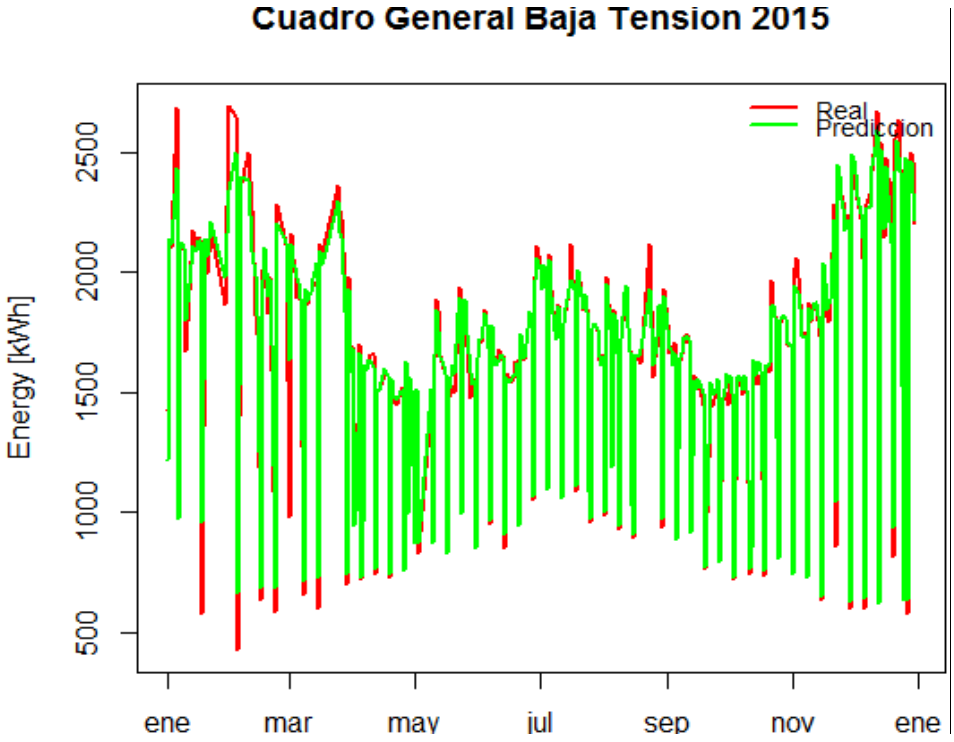
Gráfica 24: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos filtrados.



Gráfica 25: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos filtrados.

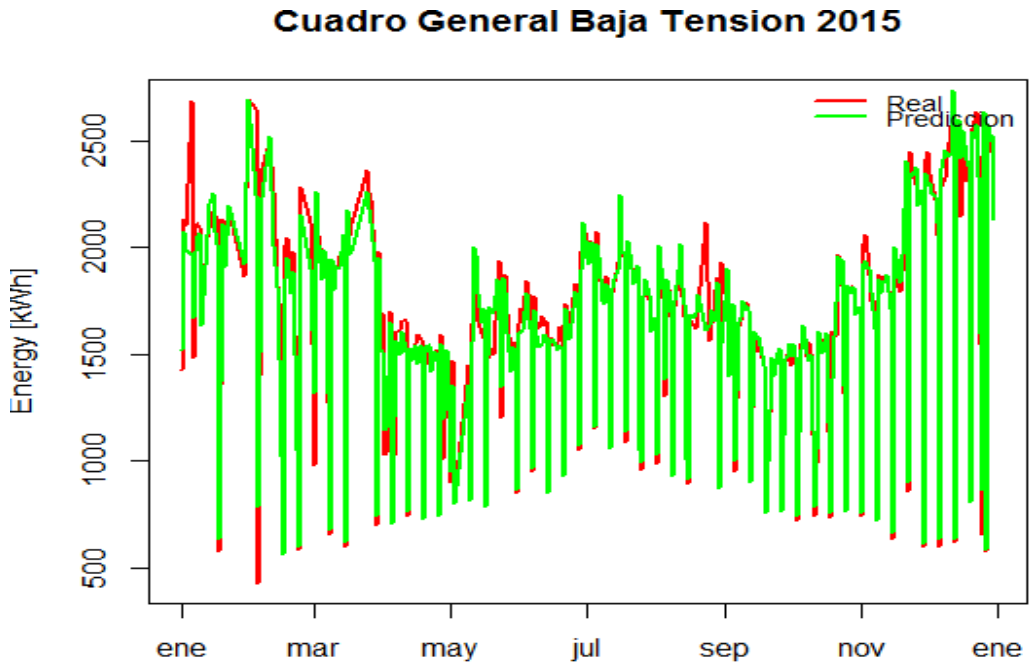


Gráfica 26: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos filtrados.

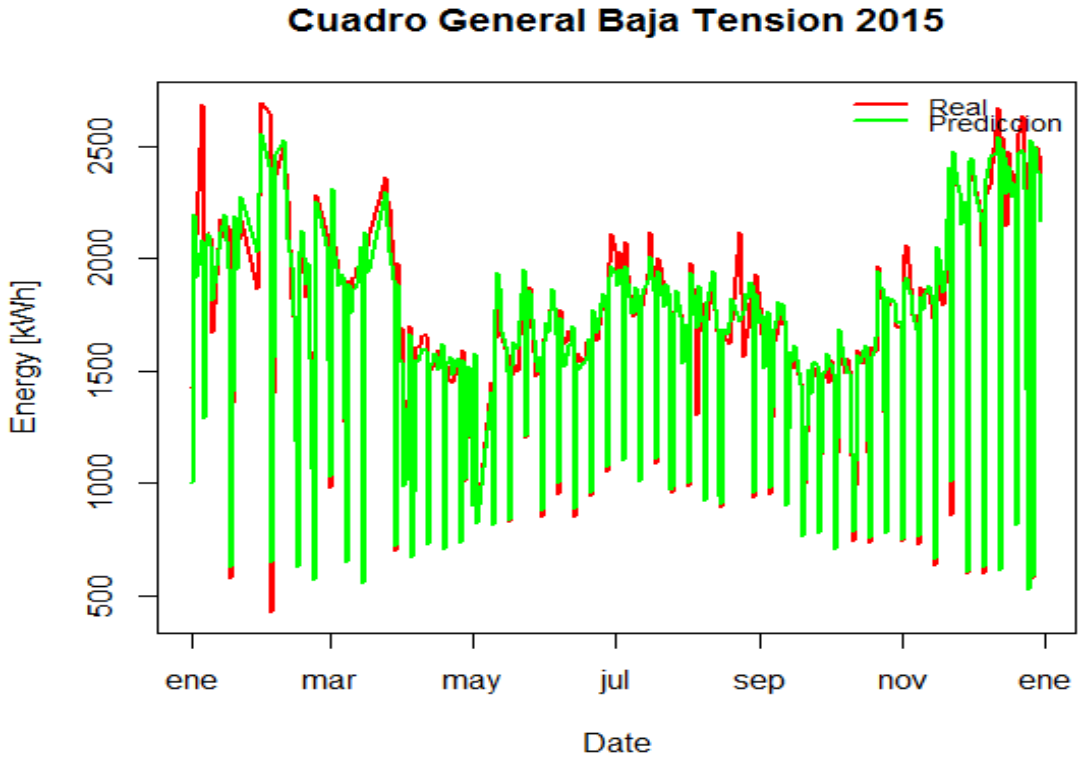


Gráfica 27: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos filtrados.

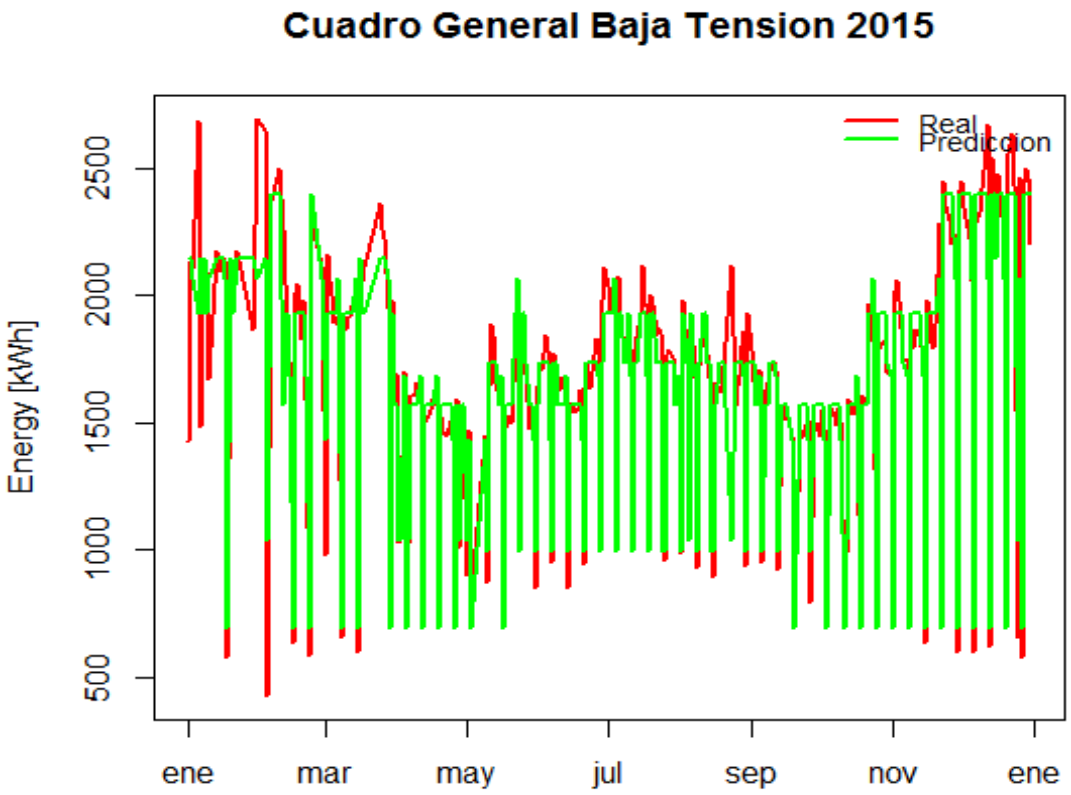
Para la versión de división de datos en subconjuntos de fines de semana y restantes;



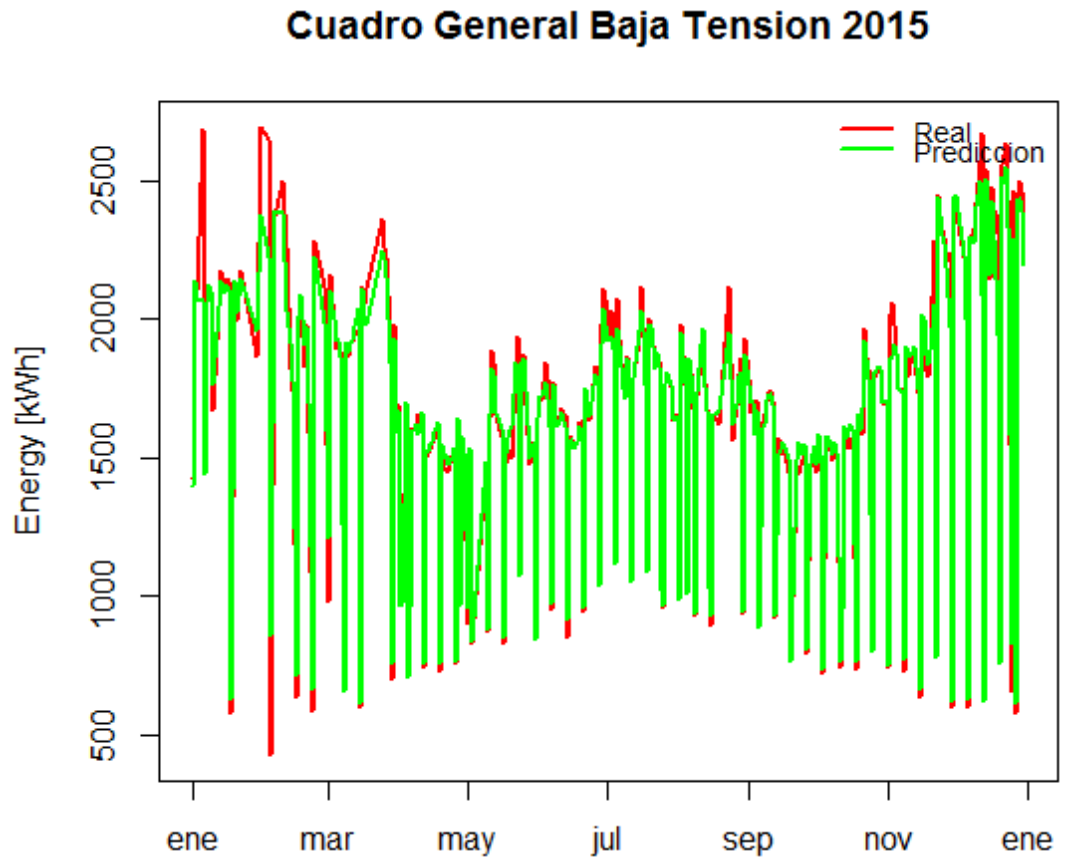
Gráfica 28: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados.



Gráfica 29: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados.

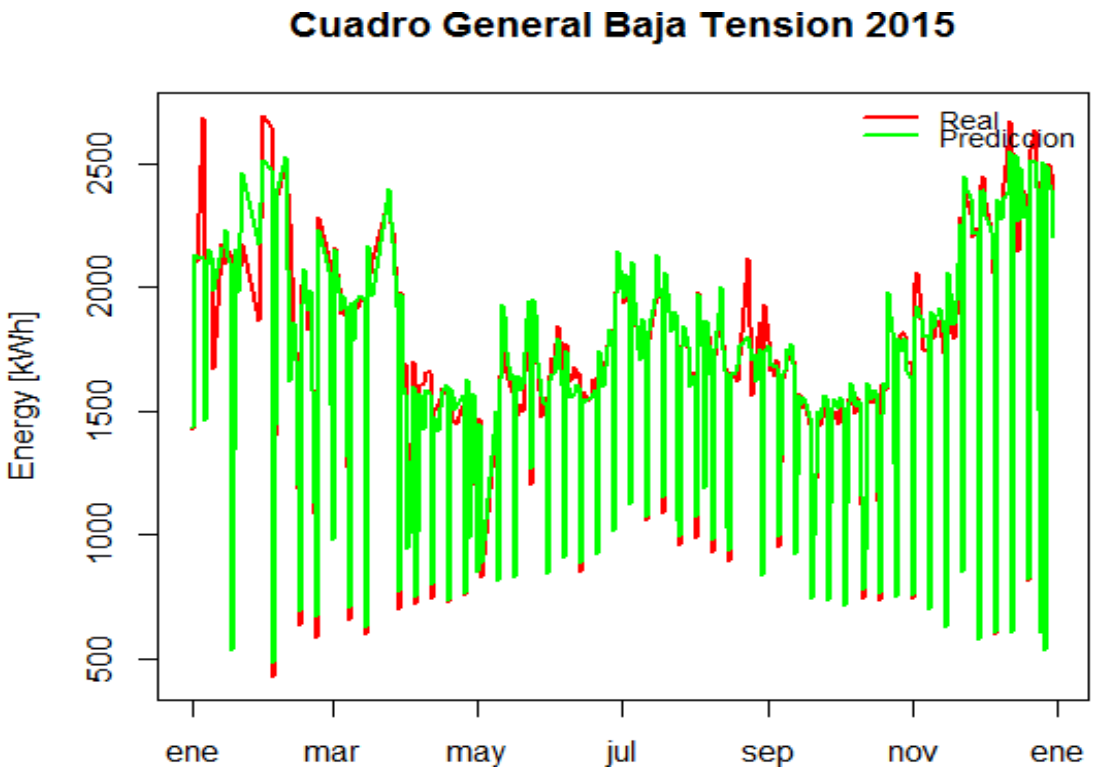


Gráfica 30: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados.

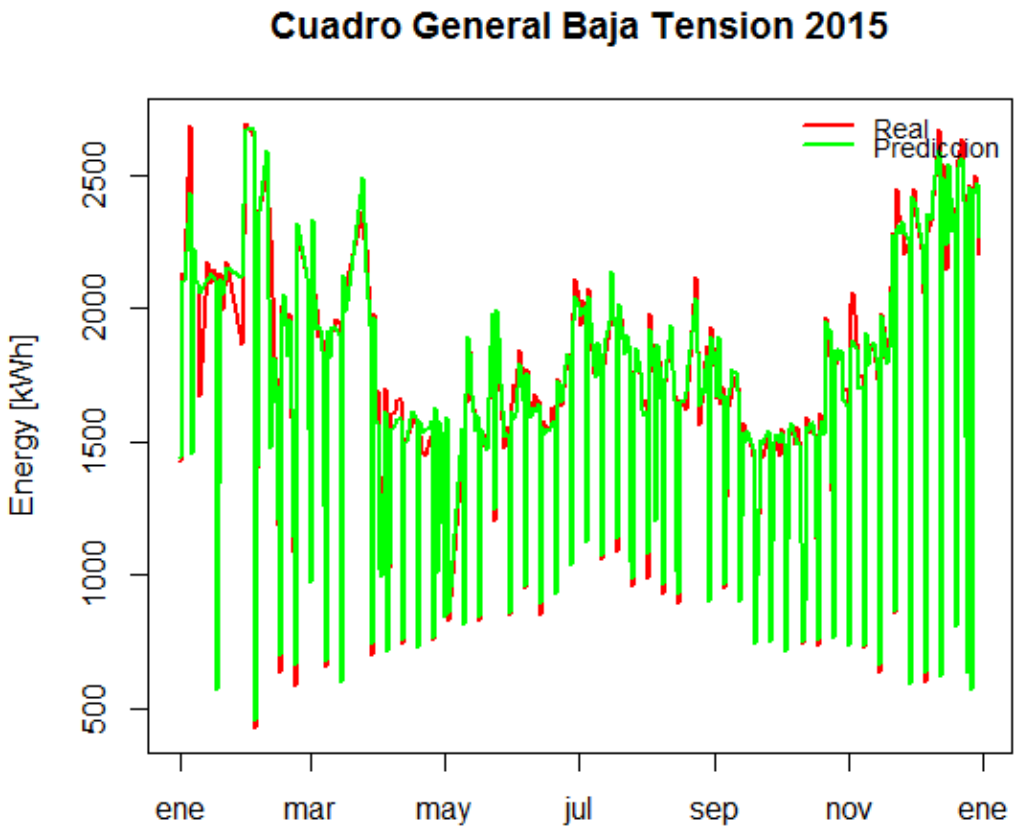


Gráfica 31: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos filtrados.

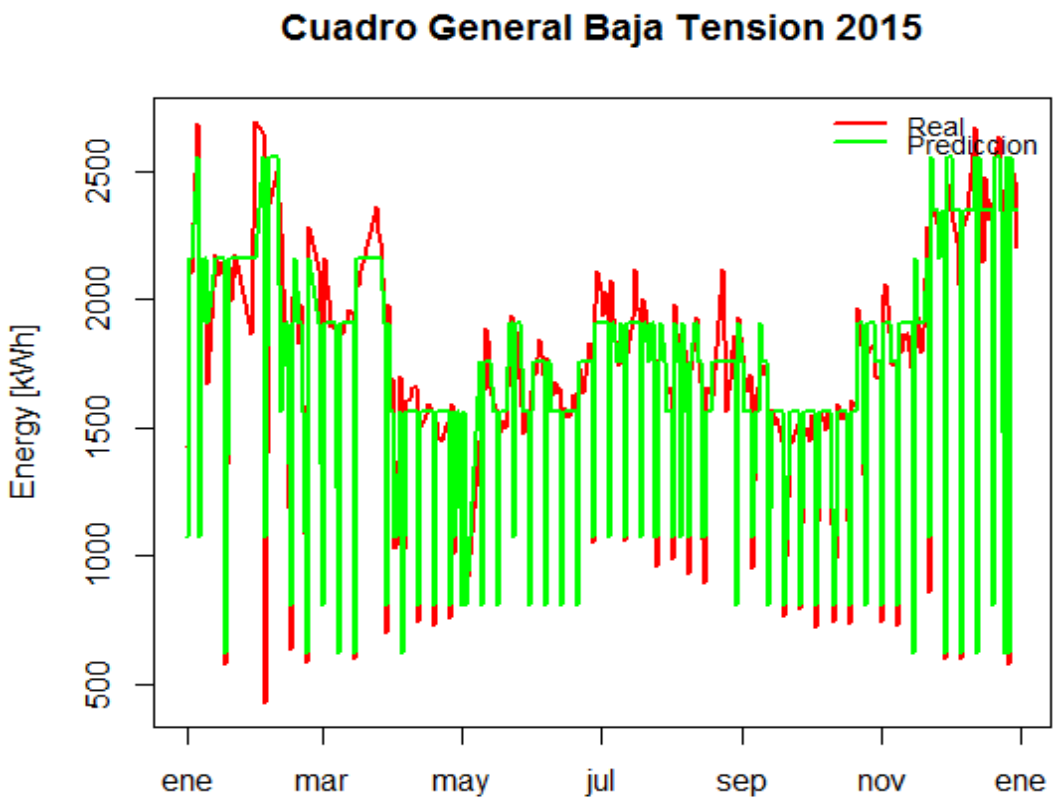
Para la versión de división de datos en subconjuntos de días laborales y festivos;



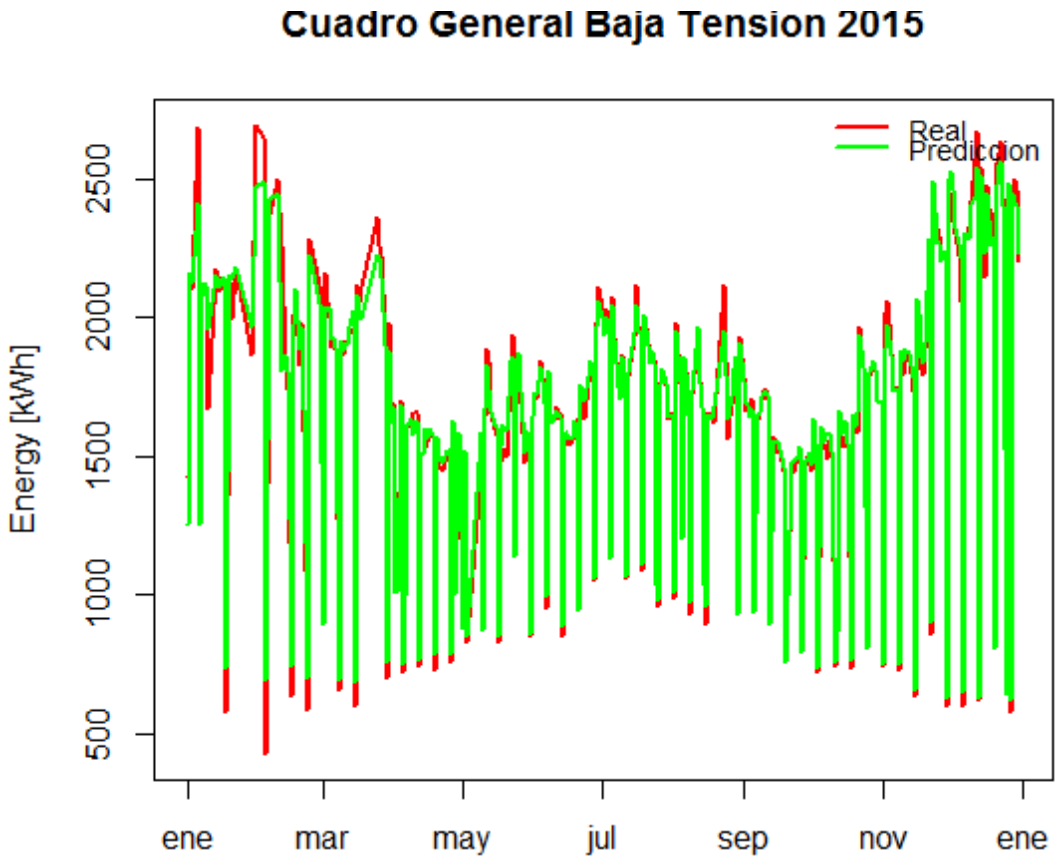
Gráfica 32: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados.



Gráfica 33: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados.

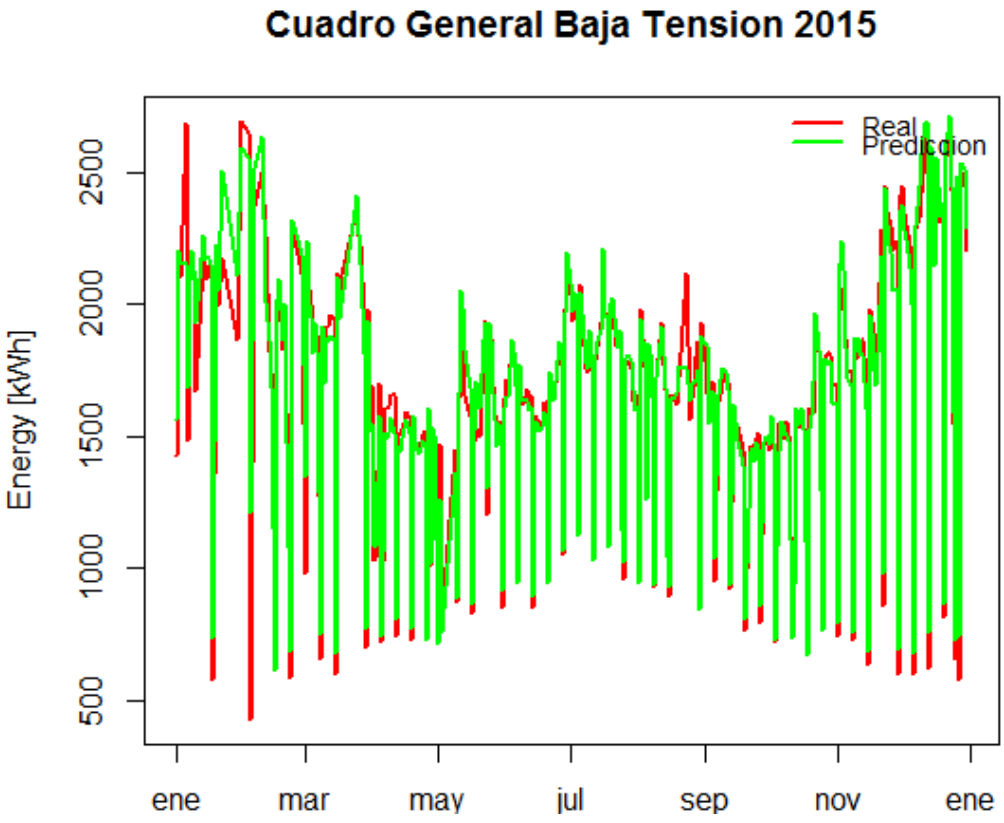


Gráfica 34: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados.

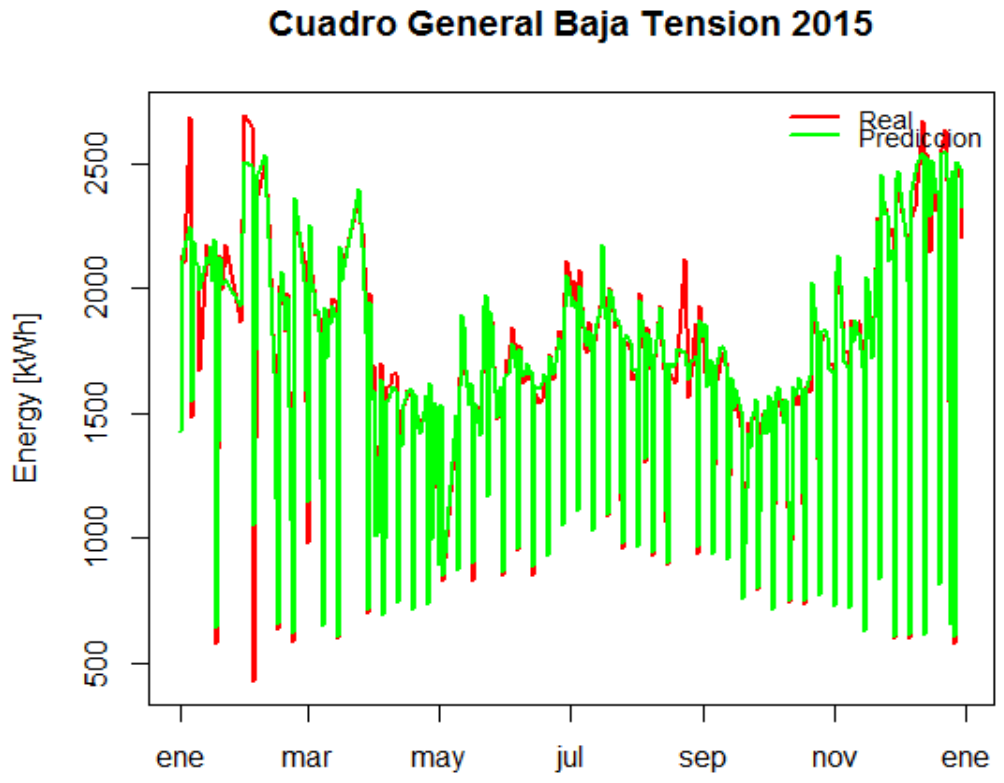


Gráfica 35: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en festivos y laborales utilizando el conjunto de datos filtrados.

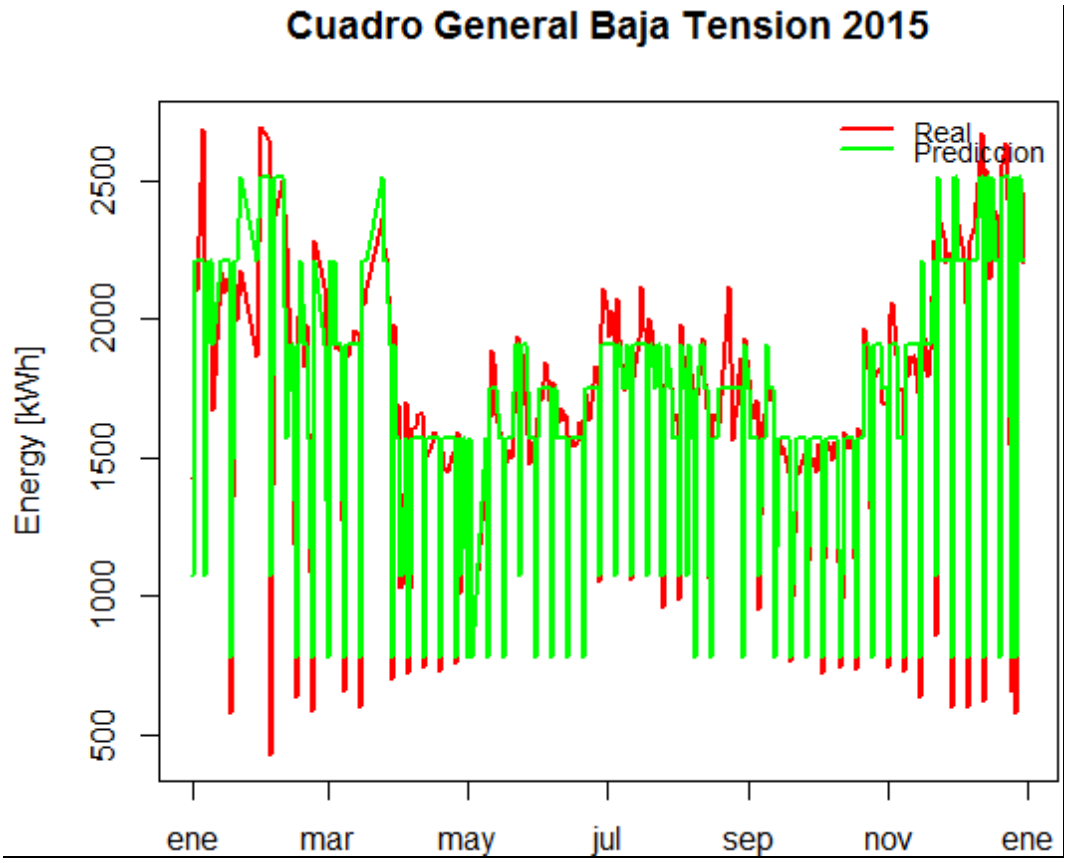
Para el conjunto de datos con el parámetro de *laboralidad* añadido;



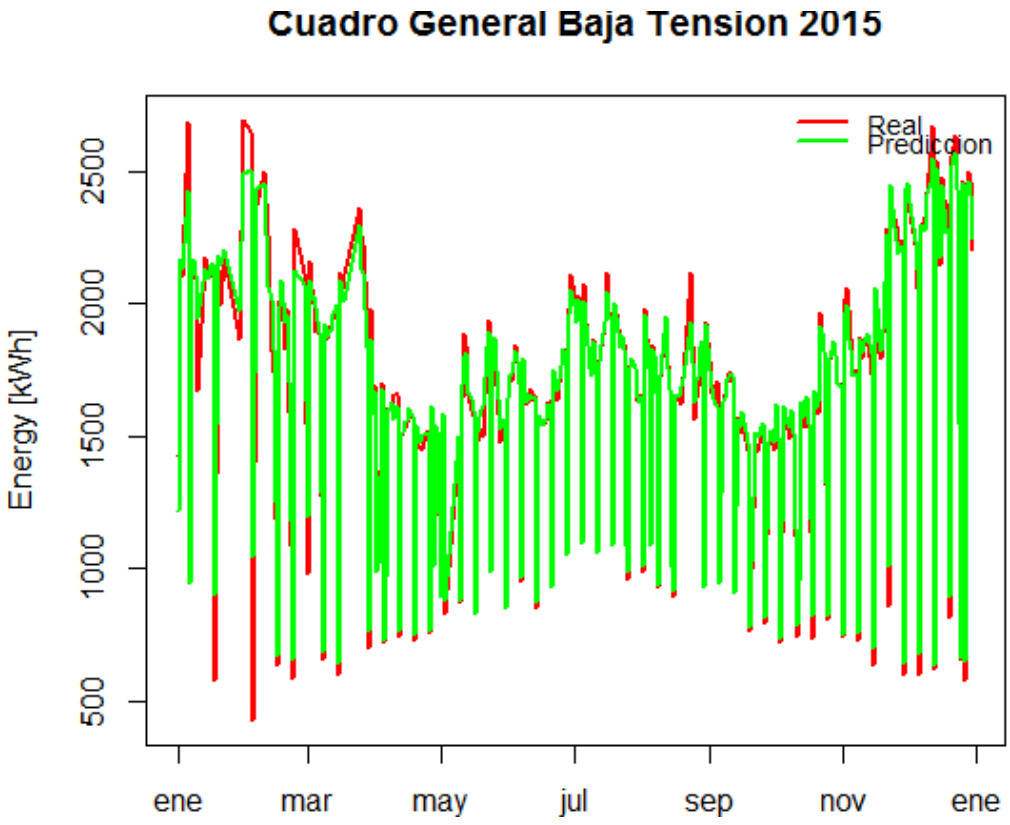
Gráfica 36: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos filtrados y añadiendo el parámetro *laboralidad*.



Gráfica 37: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad.



Gráfica 38: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad.



Gráfica 39: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos filtrados y añadiendo el parámetro laboralidad.

Los siguientes pasos son analizar y buscar una forma de intentar mejorar, si es posible, los modelos obtenidos. Para ello se deberán tener en cuenta distintas configuraciones y técnicas, cómo puede ser el uso de cross-validation o de PCA.

A continuación se aplicará la técnica de PCA para comprobar la importancia de cada variable y crear modelos con las que tengan un valor o grado mayor. Tras aplicar dicha técnica, la matriz de correlación que se obtiene es la siguiente:

Tabla 1: Matriz de valores de componentes

	PC1	PC2	PC3	PC4	PC5	PC6
exttr	0.34781471	-0.11382097	-0.06957414	0.008671801	0.03115670	0.02120615
exttemp	0.34259652	-0.06833895	0.21254133	0.069675729	-0.04201997	-0.14195577
exthabs	0.34646862	-0.11226108	-0.07609020	-0.016030954	0.10282956	-0.03485661
exthr	-0.17876604	-0.05253732	-0.67147199	-0.128673422	0.19789733	0.43667002
inttr	0.33301531	-0.14830842	-0.15961258	-0.008927225	0.09257702	0.11785180
inttemp	0.30778730	-0.12205887	0.22706977	0.051990984	0.29311076	0.49952880
inthabs	0.33238400	-0.14568817	-0.16454153	-0.038017589	0.12722462	0.03749425
inthr	0.22653124	-0.11710004	-0.56073497	-0.091162646	-0.19270472	-0.44460864
ilu1	-0.09094857	-0.42753610	0.08176468	-0.114767275	-0.07665752	-0.02623215
ilu2	-0.13504617	-0.40992590	0.04845368	-0.137798056	-0.02177180	-0.07010197
ilu3	-0.17240609	-0.37988901	0.05972270	-0.182837164	-0.06299441	-0.14143836
sv	-0.12939115	-0.41288603	0.06561178	-0.142850406	-0.05587522	-0.07352508
frio	0.27831401	-0.22844163	0.19314770	-0.038254038	0.01808509	-0.03823216
cor1	0.18388123	0.32810448	-0.01167661	-0.105425703	0.24213482	-0.44620547
cor2	-0.04444459	-0.21948495	-0.14257880	0.921241043	-0.12703314	-0.06025978
clima	-0.23665514	-0.14680586	0.01052496	0.135863978	0.84331856	-0.29966151
	PC7	PC8	PC9	PC10	PC11	PC12
exttr	-0.05325873	-0.17643049	0.364201499	-0.23212315	0.379163034	0.035975551
exttemp	0.03570737	-0.26346942	0.263927759	-0.20790425	0.186189405	0.017087461
exthabs	0.03931487	-0.14661600	0.291680414	-0.32758730	-0.625230052	0.007609464
exthr	-0.12572212	0.23045600	0.276826777	-0.18264583	0.045855122	0.017099772
inttr	-0.06893777	-0.08431071	-0.203355421	0.19846137	0.308184000	-0.025307999
inttemp	-0.24196514	-0.05371570	-0.163642540	0.25382938	0.127991116	0.013951589
inthabs	0.01306686	-0.03858921	-0.335817810	0.22967245	-0.502329288	0.004894501
inthr	0.20520903	-0.08056606	-0.198467551	0.17747165	0.196326875	0.004529435
ilu1	-0.13037429	0.08929292	-0.354601567	-0.42080852	0.049038955	0.575915433
ilu2	-0.20661347	-0.05393265	-0.118867465	-0.18702316	0.015255255	-0.783191365

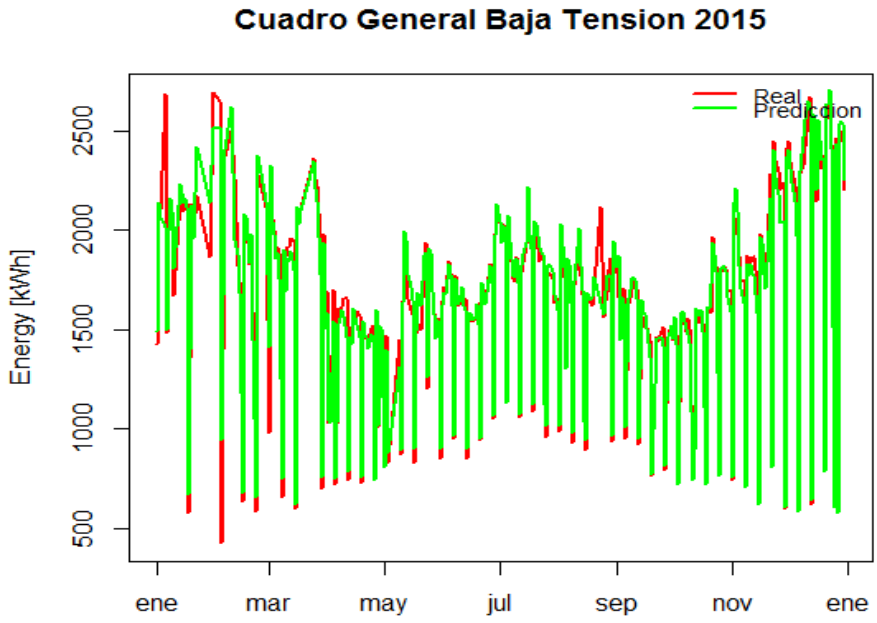
ilu3	-0.20100392	-0.06907559	0.502946850	0.59521523	-0.099361960	0.201559262
sv	-0.17652933	-0.00112678	-0.037825124	-0.06018842	-0.003650455	0.036733200
frio	0.37504290	0.80987060	0.141900289	0.03737299	0.057540368	-0.091021555
cor1	-0.69284011	0.32628990	-0.007860172	-0.04407662	0.007943834	0.019781621
cor2	-0.20483708	0.12152737	0.032935106	0.02766946	-0.049431867	-0.006279223
clima	0.28117817	-0.11873157	-0.012384800	-0.00966759	0.089299237	0.037754856
	PC13	**PC14**	**PC15**	**PC16**		
exttr	-0.526467942	0.0391326703	0.4642054599	-1,12E-09		
exttemp	0.753365067	0.1322932992	0.0491900786	-8,66E-11		
exthabs	-0.178560749	-0.1747490119	-0.4250845106	-4,98E-10		
exthr	0.288953109	0.0403464105	0.0397690109	-2,79E-10		
inttr	-0.130227846	0.5608465045	-0.5471407258	4,34E-10		
inttemp	0.080549371	-0.5693056313	-0.0291154378	-2,26E-11		
inthabs	0.100576762	0.3108496851	0.5439667896	1,04E-09		
inthr	0.074334294	-0.4599600706	-0.0477921929	-1,17E-10		
ilu1	0.003643348	-0.0104602822	-0.0088504724	-3,47E+05		
ilu2	0.006177323	-0.0237346046	0.0279538566	-2,78E+05		
ilu3	0.002424602	0.0356516047	-0.0219078191	-2,46E+05		
sv	0.004153059	-0.0017092312	-0.0007904749	8,61E+05		
frio	0.009608324	0.0011509402	-0.0052998743	5,21E-11		
cor1	0.024047004	0.0001544162	-0.0050881504	1,67E-11		
cor2	-0.002320306	-0.0011800013	0.0011986286	2,40E-11		
clima	-0.007446730	0.0020302835	0.0049826098	-1,14E-10		

Tabla 2: Importancia de las componentes

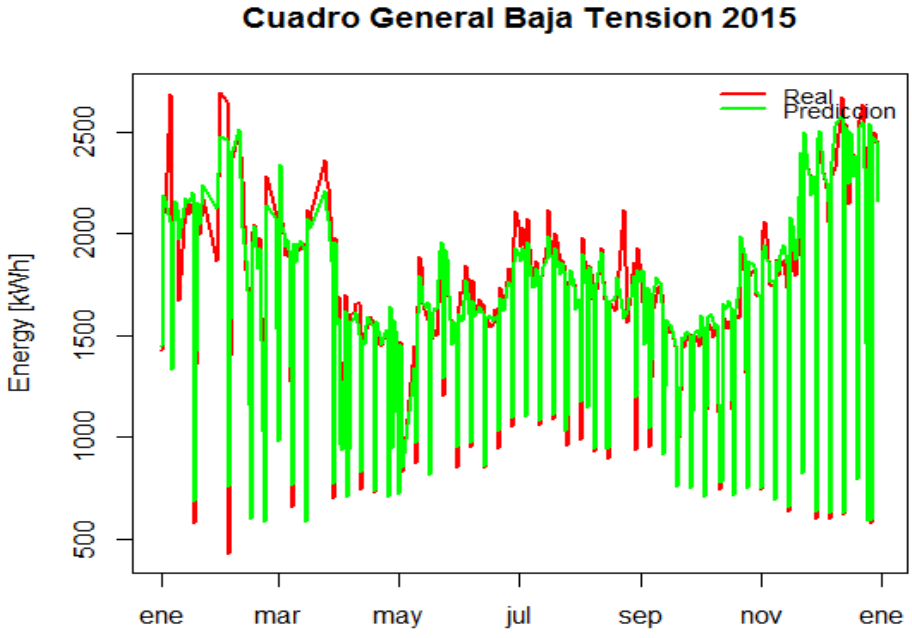
	PC1	PC2	PC3	PC4	PC5	PC6
Desviación estándar	2.754	22.196	119.590	0.90655	0.75992	0.51767
Proporción de varianza	0.474	0.3079	0.08939	0.05136	0.03609	0.01675
Porcentaje acumulado	0.474	0.7819	0.87132	0.92269	0.95878	0.97553
	PC7	**PC8**	**PC9**	**PC10**	**PC11**	**PC12**
Desviación estándar	0.46866	0.32380	0.16088	0.14165	0.11846	0.06372
Proporción de varianza	0.01373	0.00655	0.00162	0.00125	0.00088	0.00025
Porcentaje acumulado	0.98926	0.99581	0.99743	0.99868	0.99956	0.99981
	PC13	**PC14**	**PC15**	**PC16**		
Desviación estándar	0.04327	0.02968	0.01653	2,96E-13		
Proporción de varianza	0.00012	0.00006	0.00002	0.000e+00		
Porcentaje acumulado	0.99993	0.99998	1	1,00		

Cómo se puede apreciar, con las primeras 5 nuevas componentes se dispone del 95,8% de la variabilidad original y con las 8 primeras prácticamente con la totalidad de la variabilidad. Estos nuevos factores corresponden a los datos filtrados.

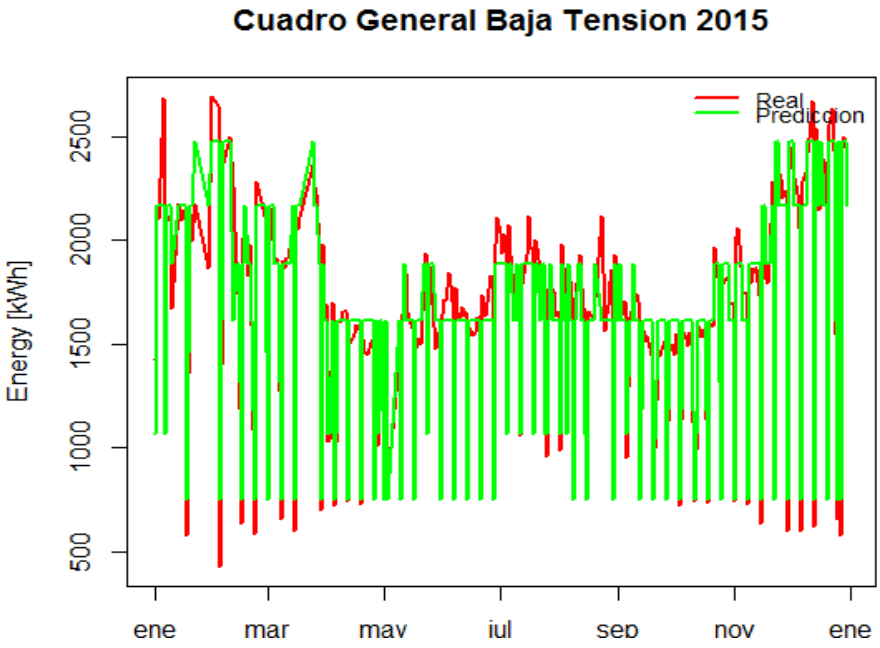
A continuación se muestran las gráficas obtenidas. A simple vista no se aprecian cambios significativos, pero posteriormente compararemos los resultados de los estimadores calculados para poder ver las diferencias.



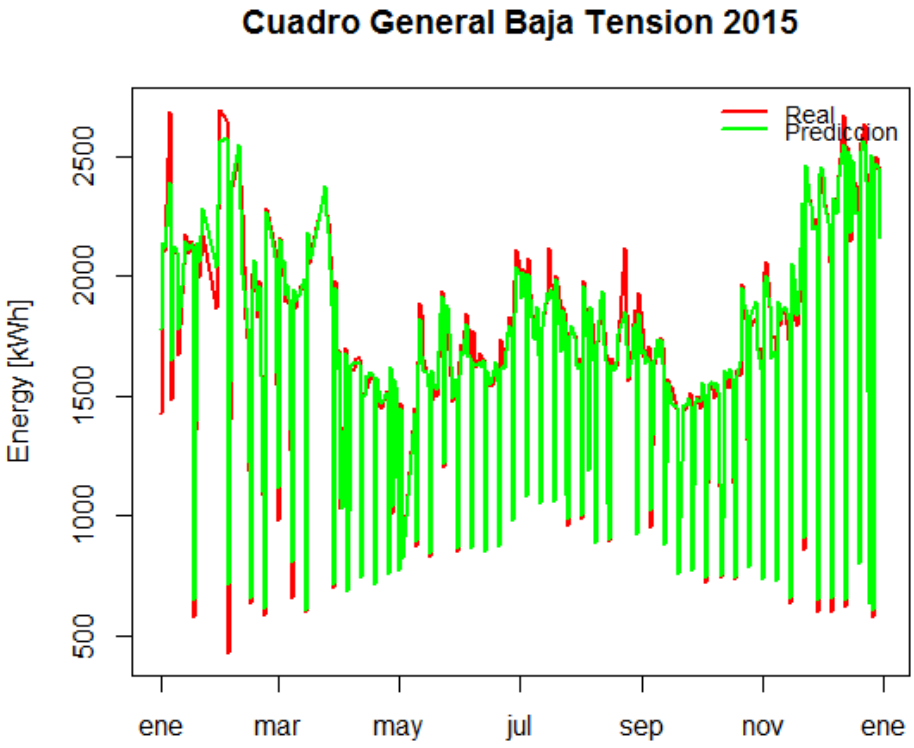
Gráfica 40: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando la técnica de PCA.



Gráfica 41: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando la técnica de PCA.



Gráfica 42: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando la técnica de PCA.



Gráfica 43: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando la técnica de PCA.

En segundo lugar se aplicará un entrenamiento y testeo con subconjuntos formados mediante validación cruzada. La razón principal por la que este método puede proporcionarnos resultados más favorables y sobretodo más fiables, es por el modo en que funciona, ya que nos asegura que se tratarán todos los datos de los que se dispone. Esta técnica se aplica directamente a los datos filtrados, ya que cómo hemos podido comprobar anteriormente, se obtienen resultados más favorables con este segundo tipo de modelos.

Esta técnica la hemos llevado a cabo de la siguiente forma: se divide el conjunto de datos en 3 subconjuntos menores, de manera que se entrena cada vez con dos de las tres particiones y se testea con la restante, para finalmente realizar un promedio. Las gráficas obtenidas mediante la validación cruzada no se muestran debido a que realmente no se pueden distinguir visualmente por ejemplo de las vistas anteriormente, por lo que para observar si verdaderamente producen mejores resultados se tendrán en cuenta los estimadores obtenidos que veremos posteriormente.

4. MARCO EXPERIMENTAL

En esta sección se va a explicar el marco experimental tenido en cuenta en las ejecuciones de este proyecto. Se explicará en detalle todos los parámetros, datos, medidas de error… que se han tenido en cuenta.

Debido a la gran cantidad de parámetros y funciones tenidos en cuenta en el proyecto se debe intentar explicar el significado de los más importantes:

- **Parámetros del sistema:**
  - raíz: variable que guarda el directorio de trabajo.
  - Set.seed(): se utiliza para fijar una semilla utilizada para generar los números aleatorios a lo largo de la ejecución y que los resultados no se vean afectados por la variación de distintas semillas. El valor escogido es el 13310.
  - Source("funciones.R"): incluye el archivo "funciones.R" que contiene varias funciones para la creación del modelo de datos, normalización, aplicación de técnicas de predicción, cálculo de errores y representación de resultados.

- **Datos de entrada:**
  - dataint: contiene los datos del archivo csv correspondiente a las variables del interior del supermercado.
  - dataext: contiene los datos del archivo csv correspondiente a las variables del exterior del supermercado.
  - datailu1, datailu2, datailu3: contienen los datos de los archivos correspondientes a las variables de consumo por alumbrado.
  - datafrio1, datafrio2: contienen los datos de los archivos correspondientes a las variables de consumo por frio positivo y frio negativo.
  - cgbt: contiene los datos del archivo correspondiente a las variables de consumo general de la tienda.
  - datacor1, datacor2: contienen los datos de los archivos correspondientes a las variables de consumo por cortina.
  - dataclima1, dataclima2: contienen los datos de los archivos correspondientes a las variables de consumo del tejado.
  - tempoficial: contiene los datos del archivo csv correspondiente a la temperatura exterior (en Vitoria) proporcionada por EUSKALMET.

- **Datos del modelo:**
  - total_dia: agrupa cada una de las variables anteriores en una sola, de manera que en cada columna tengamos una variable objeto de estudio (ej: humedad interior), y las agrupa por días.
  - modelo: tiene la información de "total_dia" en el formato adecuado para poder trabajar con las funciones que se utilizarán posteriormente en la predicción.
    Dependiendo de la ejecución que se realiza, la variable contendrá unos datos u otros. La función "crearModelo" recibe como parámetro de entrada "tipo", un

número que va de 0 a 3, “outliers”, que puede ser 0 o 1, y “datos”, la variable total_dia con los datos que se van a tratar, y devuelve la variable con el formato adecuado y los datos que se haya seleccionado.

- Tipo:
  - Valor 0: se creará con los datos para los días de la semana de lunes a viernes.
  - Valor 1: se creará con los datos para los sábados y domingos.
  - Valor 2: se creará con los datos para los días laborales.
  - Valor 3: se creará con los datos para los días festivos.
  - Valor 4: a cada uno de los datos se le añade una nueva variable que indica su “laboralidad”.
- Outliers:
  - Valor 0: no se eliminan los valores anómalos de la variable.
  - Valor 1: los valores anómalos son eliminados.

- datsModelo: Esta variable es el resultado de la llamada a la función crearDatos(modelo). Contiene una lista con los datos necesarios para la creación posterior de los conjuntos “train” y “test” y normalizar los datos.
- datsNormalizados: Contiene los conjuntos “train”, “test” y “scaled” (todos los datos normalizados). Se obtiene de la función normalizar(modelo, tipo, datsModelo).
  - Tipo:
    - Valor 1: Los datos se normalizan mediante la función max-min. { (x - min(x))/(max(x) - min(x)) }. Los resultados estarán en el rango [0, 1].
    - Valor 2: Normalización similar a la de max-min pero que se utiliza cuando se desean unos datos más centralizados. { x-((max+min)/2) / (max-min)/2 }
    - Otro valor: Normalización estándar o z-score.

- red: contiene los resultados obtenidos tras entrenar el modelo con una red neuronal y aplicar el conocimiento al conjunto “test”. Esto es, contiene los valores que predice la red neuronal. Esta variable se obtiene tras la llamada a la función “entrenamientoRed”.
- regresión: contiene los resultados obtenidos tras entrenar el modelo con una regresión lineal y aplicar el conocimiento al conjunto “test”. Esto es, contiene los valores que predice la regresión. Esta variable se obtiene tras la llamada a la función “entrenamientoLM”.
- arbol: contiene los resultados obtenidos tras entrenar el modelo con un árbol de regresión y aplicar el conocimiento al conjunto “test”. Esto es, contiene los valores que predice el árbol. Esta variable se obtiene tras la llamada a la función “entrenamientoArbol”.
- randomf: contiene los resultados obtenidos tras entrenar el modelo con un random forest y aplicar el conocimiento al conjunto “test”. Esto es, contiene

los valores que predice el random forest. Esta variable se obtiene tras la llamada a la función "entrenamientoForest".

- errors: contiene los estimadores MSE, Rsquared, MAPE y EME para cada una de las técnicas anteriores. Se obtiene tras la llamada a la función estimadores(real, regresion, red, arbol, randomf), dónde real son los valores para la variable CGBT reales que se quieren predecir y en este caso, estimar.

- **Otras:**
  - Se utilizan además varias variables auxiliares y que tienen como finalidad el conseguir proporcionar ayuda para la construcción del código mediante creación de nuevas variables, lecturas de datos, bucles y sentencias de recorrido de datos, etc.

5. RESULTADOS

En esta sección se verán las tablas con los estimadores calculados. Recordamos que los que se calculan son los siguientes: Error cuadrático medio (ECM), Error estándar o raíz del error cuadrático medio (RECM), *Mean Absolute Percentage Error* (MAPE) y Error medio de energía (EME). También incluimos el valor *Rsquared* o R cuadrado, un coeficiente que determina la calidad y proporción de variación del modelo. Las tablas de resultados siguen la siguiente nomenclatura:

- **Versión i.j:** Dónde el parámetro *i* hace referencia a los datos utilizados; 0 si son los datos originales y 1 si se han filtrado, es decir, si se han tenido en cuenta las técnicas de eliminación de datos erróneos, detección de *outliers,* etc. El parámetro *j* se refiere a la clasificación de los datos; 0 si no hay clasificación, 1 si se dividen en fines de semana y restantes, 2 si se dividen en laborales y festivos y 3 si no se dividen pero se añade el parámetro denominado "laboralidad".

Así pues, tenemos las siguientes tablas de resultados:

Tabla 3: Train versión 1.0

	Version 1.0				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	61,67790259	0,769316874	23,21242	11,4652	7,853528
Red Neuronal	25,34945069	0,905189861	9,498839	5,045452	5,034824
Arbol decision	94,00382534	0,64841385	15,08952	9,488162	9,695557
Random Forest	42,66679967	0,840420794	7,840446	4,494506	6,531983

Tabla 4: Test versión 1.0

	Version 1.0				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	121,5954	0,651614	19,14458	8,835256	11,02703
Red Neuronal	63,68725	0,817528	6,434311	6,238155	7,980429
Arbol decision	177,537	0,491335	26,74982	12,66334	13,3243
Random Forest	118,2408	0,661226	17,43428	8,443776	10,87386

Tabla 5: Train versión 1.1

	Version 1.1				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	53,89073442	0,798441864	17,04848791	10,37017153	7,341031
Step Regresion	59,38983	0,7778745	12,84285	6,759782	7,70648
Red Neuronal	31,68661643	0,881488062	17,27843261	6,173888546	5,629087
Arbol decision	79,65392209	0,702084296	21,53378965	10,82025648	8,924905
Random Forest	29,93403918	0,888042922	15,03461279	5,38798223	5,471201

Tabla 6: Test versión 1.1

	Version 1.1				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	39,03722	0,818337	31,40306	9,290364	6,247977
Red Neuronal	62,21032	0,710498	50,45659	9,575442	7,887352
Arbol decision	68,13228	0,68294	50,4474	11,35212	8,254228
Random Forest	48,81026	0,772857	55,03049	8,912652	6,986434

Tabla 7: Train versión 1.2

	Version 1.2				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	52,12352364	0,805051455	11,36481	9,893076	7,219662
Step Regresion	59,99562	0,7756088	8,675168	7,483852	7,745684
Red Neuronal	16,15510768	0,939577862	7,408787	5,470062	4,019342
Arbol decision	68,15942056	0,74507518	16,54319	9,171716	8,255872
Random Forest	19,95578932	0,925362834	8,92359	4,509834	4,46719

Tabla 8: Test versión 1.2

	Version 1.2				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	76,77995	0,63726	12,37139	11,2136	8,762417
Red Neuronal	52,47884	0,752069	6,621406	7,013182	7,244228
Arbol decision	97,29654	0,540332	13,18803	12,19953	9,863901
Random Forest	30,61775	0,855349	7,412787	6,740283	5,533331

Tabla 9: Train versión 1.3

	Version 1.3				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	70,17586	0,737533	19,12746	7,655041	8,377104
Red Neuronal	80,5609	0,698692	15,76666	8,864061	8,975572
Arbol decision	100,3376	0,624725	18,81315	9,781655	10,01687
Random Forest	62,53079	0,766127	8,499141	5,17211	7,907641

Tabla 10: Test versión 1.3

	Version 1.3				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	125,4007	0,640712	18,21467	9,244138	11,19824
Red Neuronal	62,63906	0,820532	9,416936	6,186299	7,914484
Arbol decision	177,537	0,491335	26,74982	12,66334	13,3243
Random Forest	120,302	0,65532	17,11999	8,257284	10,96823

Tabla 11: Train versión 2.0

	Version 2.0				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	5,257054	0,961384	4,100174	3,411837	2,292827
Red Neuronal	1,811232	0,986696	2,543074	2,03568	1,34582
Arbol decision	9,326041	0,931495	6,074597	5,157686	3,053857
Random Forest	3,216092	0,976376	2,878922	2,408483	1,793347

Tabla 12: Test versión 2.0

	Version 2.0				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	6,894763	0,947817	3,309091	3,043286	2,625788
Red Neuronal	6,170214	0,953301	3,467458	3,237545	2,483992
Arbol decision	7,603737	0,942451	4,998048	4,65067	2,757487
Random Forest	7,650041	0,942101	4,849385	4,236248	2,765871

Tabla 13: Train versión 2.1

	Version 2.1				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	6,19193885	0,954517036	4,219634403	3,929436111	2,488361
Step Regresion	4,605861	0,9661676	3,414959	3,084749	2,146127
Red Neuronal	4,539196361	0,966657276	3,677175682	3,500445655	2,130539
Arbol decision	14,98182196	0,88995084	6,628135876	5,879760193	3,870636
Random Forest	4,679441575	0,965627103	3,12737725	2,848895736	2,163202

Tabla 14: Test versión 2.1

	Version 2.1				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	11,34278	0,925172	5,569291	4,191264	3,367905
Red Neuronal	18,05172	0,880913	6,800687	5,164266	4,248731
Arbol decision	40,67065	0,731697	11,45166	8,394507	6,377354
Random Forest	23,76238	0,84324	8,923599	6,862845	4,874667

Tabla 15: Train versión 2.2

	Version 2.2				
	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	3,927946731	0,971147218	3,178339	3,016955	1,981905
Step Regresion	3,594611	0,9735957	2,576752	2,41974	1,895946
Red Neuronal	2,980695556	0,978105264	2,484054	2,398908	1,726469
Arbol decision	8,060831971	0,940789058	5,551796	4,888537	2,83916

Random Forest	2,422823956	0,982203116	2,756991	2,40662	1,556542

Tabla 16: Test versión 2.2

	Version 2.2				
	MSE	**Rsquared**	**MAPE**	**EME**	**RMSE**
Regresion Lineal	4,164761	0,972819	3,431009	3,316662	2,040775
Red Neuronal	2,56123	0,983285	2,875197	2,461647	1,600384
Arbol decision	11,1586	0,927176	6,32446	5,808389	3,34045
Random Forest	8,062766	0,94738	4,612186	4,451497	2,839501

Tabla 17: Train versión 2.3

	Version 2.3				
	MSE	**Rsquared**	**MAPE**	**EME**	**RMSE**
Regresion Lineal	7,403584	0,945617	5,439248	4,532001	2,720953
Red Neuronal	3,621563	0,973398	3,327336	2,996893	1,90304
Arbol decision	8,785217	0,935468	6,228767	5,294999	2,963987
Random Forest	3,703075	0,972799	3,205914	2,70119	1,924338

Tabla 18: Test versión 2.3

	Version 2.3				
	MSE	**Rsquared**	**MAPE**	**EME**	**RMSE**
Regresion Lineal	7,911608	0,939227	4,052331	3,725457	2,812758
Red Neuronal	6,208278	0,952311	4,114272	3,534741	2,491642
Arbol decision	9,775646	0,924909	6,408176	5,624596	3,126603
Random Forest	7,212558	0,944597	4,289269	4,010601	2,685621

Tabla 19: Comparación entre los mejores de cada versión.

	Mejores				
	MSE	**Rsquared**	**MAPE**	**EME**	**RMSE**
Version 1.0	63,68725	0,817528	6,434311	6,238155	7,980429
Version 1.1	39,03722	0,818337	31,40306	9,290364	6,247977
Version 1.2	30,61775	0,855349	7,412787	6,740283	5,533331
Version 1.3	62,63906	0,820532	9,416936	6,186299	7,914484
Version 2.0	6,170214	0,953301	3,467458	3,237545	2,483992
Version 2.1	11,34278	0,925172	5,569291	4,191264	3,367905
Version 2.2	2,56123	0,983285	2,875197	2,461647	1,600384
Version 2.3	6,208278	0,952311	4,114272	3,534741	2,491642

A la vista de los resultados, podemos afirmar que la mejor técnica resulta para los datos pre procesados, utilizando la clasificación por días laborales y festivos y aplicando la técnica de red neuronal. Puesto que es claramente superior en cuanto a resultados al resto de opciones,

únicamente se aplicará a la opción vencedora y a la segunda mejor las técnicas de validación cruzada y PCA para ver si los resultados mejoran o se mantienen similares.

Así, utilizando esos dos modelos pero con las variables obtenidas tras el PCA, los resultados obtenidos son:

Tabla 20: Train versión 2.3 y técnica PCA.

Version 2.3	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	5,106068485	0,96249331	3,832514294	3,44472	2,259661
Red Neuronal	4,92199478	0,963845426	4,093222272	3,670522	2,218557
Arbol decision	9,545285602	0,929884985	6,601866725	5,752049	3,089545
Random Forest	2,227724313	0,983636223	2,668475753	2,341406	1,492556

Tabla 21: Test versión 2.3 y técnica PCA.

Version 2.3	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	2,94966	0,97614	3,409144	3,119306	1,717458
Red Neuronal	2,25389	0,981768	2,88067	2,742414	1,501296
Arbol decision	9,63829	0,922035	6,900389	6,017824	3,10456
Random Forest	3,53833	0,971378	3,649322	3,275798	1,881044

Tabla 22: Train versión 2.2 y técnica PCA

Version 2.2	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	7,19563	0,947144	5,099665	4,448254	2,682467
Red Neuronal	3,85035	0,971717	3,65128	3,39796	1,96223
Arbol decision	8,60283	0,936808	6,094703	5,158091	2,933058
Random Forest	2,38906	0,982451	2,677048	2,342803	1,545658

Tabla 23: Test versión 2.2 y técnica PCA.

Version 2.2	MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	3,07012	0,978281	2,970139	2,907552	1,752175
Red Neuronal	9,16677	0,93515	3,613668	3,645838	3,027667
Arbol decision	15,1006	0,893171	6,891012	6,123984	3,885944
Random Forest	7,85684	0,944417	4,536006	4,314474	2,803005

Aplicando validación cruzada con 3 subconjuntos, los resultados obtenidos son:

Tabla 24: Train versión 2.3 y técnica de validación cruzada con K = 3

	Version 2.3 MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	4,69407	0,975412	3,563313	3,379076	2,16658
Red Neuronal	3,57317	0,97754	3,152031	2,83476	1,890283
Arbol decision	10,3526	0,943214	6,243359	5,602656	3,217546
Random Forest	6,61591	0,968874	4,239654	3,789895	2,572141

Tabla 25: Test versión 2.3 y técnica de validación cruzada con K = 3.

	Version 2.3 MSE	Rsquared	MAPE	EME	RMSE
Regresion Lineal	6,56807	0,951754	4,376013	3,857076	2,562824
Red Neuronal	5,44717	0,952791	3,964731	3,31276	2,333916
Arbol decision	12,2266	0,910189	7,056059	6,080656	3,496662
Random Forest	8,48991	0,937637	5,052354	4,267895	2,913745

6. CONCLUSIONES

EL objetivo de este proyecto era el de realizar un estudio del comportamiento de varias técnicas de inteligencia artificial, aprendizaje automático… para unos datos de un proyecto concreto llevado a cabo por CENER, con el objetivo de conocer y comprender el funcionamiento de dichas técnicas así como utilizar los resultados de las mismas para poder calcular ahorros, realizar simulaciones de datos futuros, etc.

Así pues, hemos visto las técnicas empleadas y las distintas configuraciones y combinaciones, y podemos concluir que por lo menos un par de modelos de los que se han construido y empleado (Red neuronal - Versión 2.3 aplicando PCA y Red neuronal – Versión 2.2) son capaces de predecir el valor del consumo de CGBT dependiendo de las distintas variables analizadas de manera correcta, esto es, con un error bajo, y que por lo tanto se pueden emplear para predecir futuras instancias. También observamos que las redes neuronales son una técnica que funciona muy bien y que es muy precisa.

El modelo que mejor se comporta de ambos y que por lo tanto será el elegido para llevar a cabo las predicciones es el modelo de Red neuronal - Versión 2.3, es decir, para los datos pre procesados y con el parámetro *laboralidad* añadido y aplicando también PCA.

Hay que tener en cuenta que hasta ahora no se han producido todas las obras de renovación del supermercado y que por lo tanto todavía no puede establecerse un ahorro concreto.

Cabe destacar también que los datos utilizados han sido para el año 2015 y que por tanto conforme avance el tiempo los modelos utilizados pueden empeorar su rendimiento. Para

evitar el incremento del error en nuevos datos, existen varias posibilidades; la primera consiste simplemente en revisar los modelos y cambiar ciertos parámetros y configuraciones para ver si consigue mejorar el rendimiento. Otra opción sería entrenar de nuevo los modelos pero esta vez utilizando más datos, incluyendo años anteriores. Aun así, esto no garantiza que dichos modelos vayan a funcionar correctamente a lo largo de los años, ya que cómo se ha dicho, la predicción del consumo de energía depende de muchos factores que varían constantemente a lo largo del tiempo como puede ser la temperatura y la humedad, por lo que siempre habrá que tener este aspecto presente a la hora de analizar los nuevos datos obtenidos.

7. BIBLIOGRAFÍA

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani; "An Introduction to Statistical Learning with Applications in R"

Ricardo Enríquez Miranda, Mª José Jiménez Taboada, Mª del Rosario Heras Celemín, "Evaluación energética experimental de edificios en condiciones reales de uso mediante el ajuste de modelos de simulación con aplicaciones al control predictivo", Madrid 2013

Max Kuhn, Kjell Johnson, "Applied Predictive Modeling"

Guillermo Escrivá Escrivá, Carlos Álvarez Bel, "Nuevas herramientas para facilitar la respuesta activa de consumidores en mercados eléctricos liberalizados: implementación y retribución", Valencia 2009

Luis Hernández Callejo, Belén Carro Martínez, Antonio Javier Sánchez Esquevillas, Javier Manuel Aguilar Pérez, "Aplicación de técnicas no lineales y otros paradigmas en *Smart grid/microgrid/virtual power plant*

Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning: Data Mining, Inference and Prediction", Second Edition

Johanny Franchesco Niño Fonseca, "Control de procesos con redes neuronales artificiales: *Metodología de diseño de control de procesos utilizando redes neuronales artificiales*", 2010

José Antonio Vázquez-López, Ismael López-Juárez, "Control Estadístico de Procesos por Redes Neuronales Artificiales: *Uso de la FuzzyARTMAP para el Reconocimiento de Patrones en Gráficos de Control*"

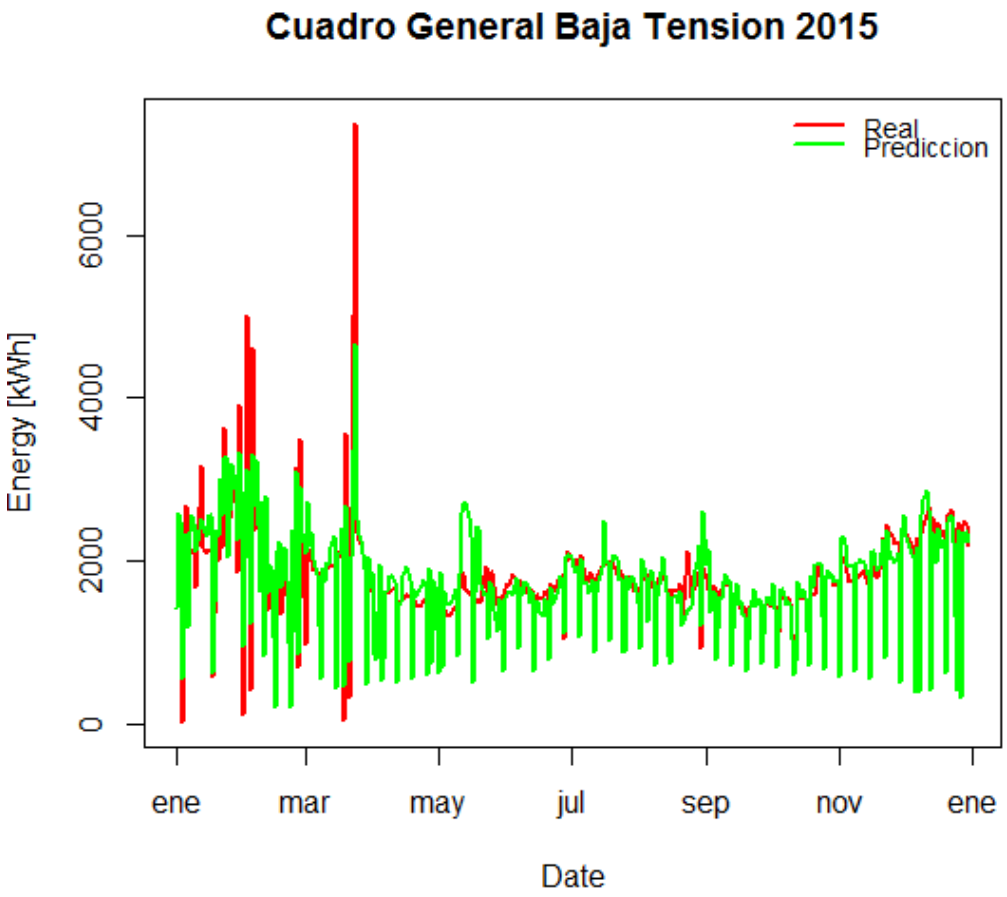
ASHRAE GUIDELINE, "Measurement of Energy and Demand Savings", American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

Fahad A. Al-Sulaiman, Ibrahim Dincer, Feridun Hamdullahpur, "Energy and exergy of a biomass trigeneration system using an organic Rankine cycle"

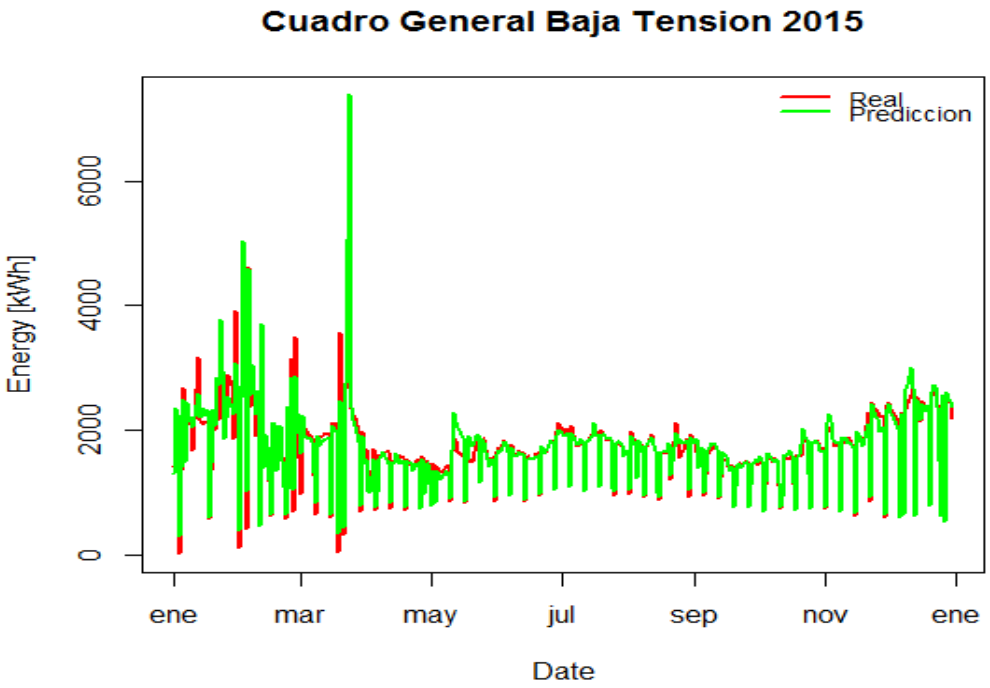
8. ANEXO

8.1 ANEXO 1

Gráficas obtenidas tras la aplicación de varios modelos utilizando el conjunto de datos original, sin realizar ningún pre procesado.

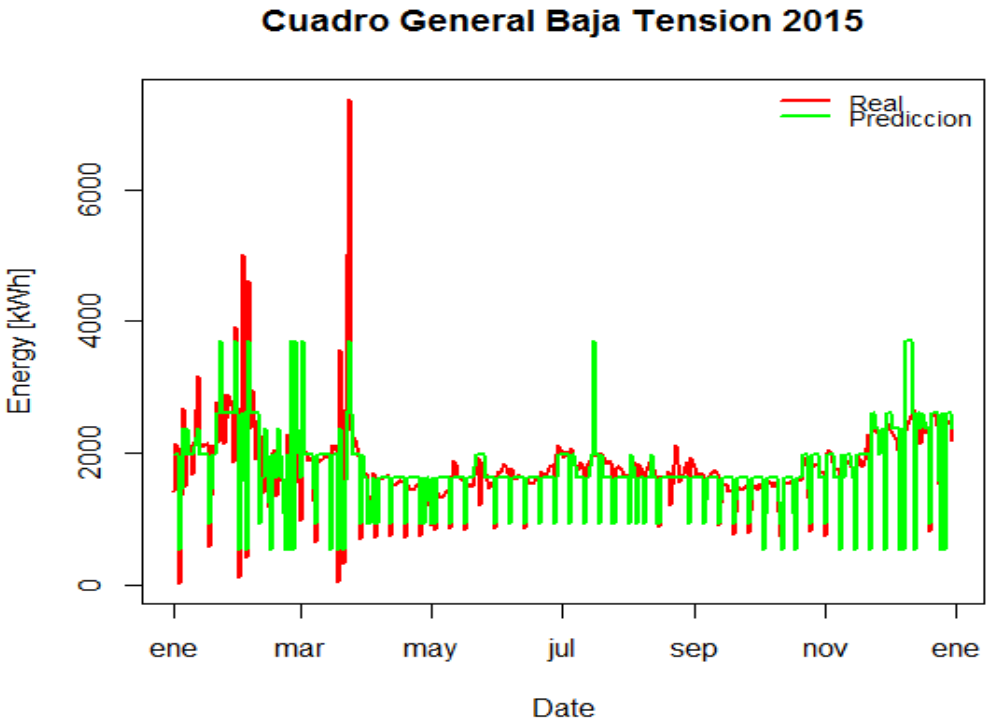


Gráfica 44: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales.



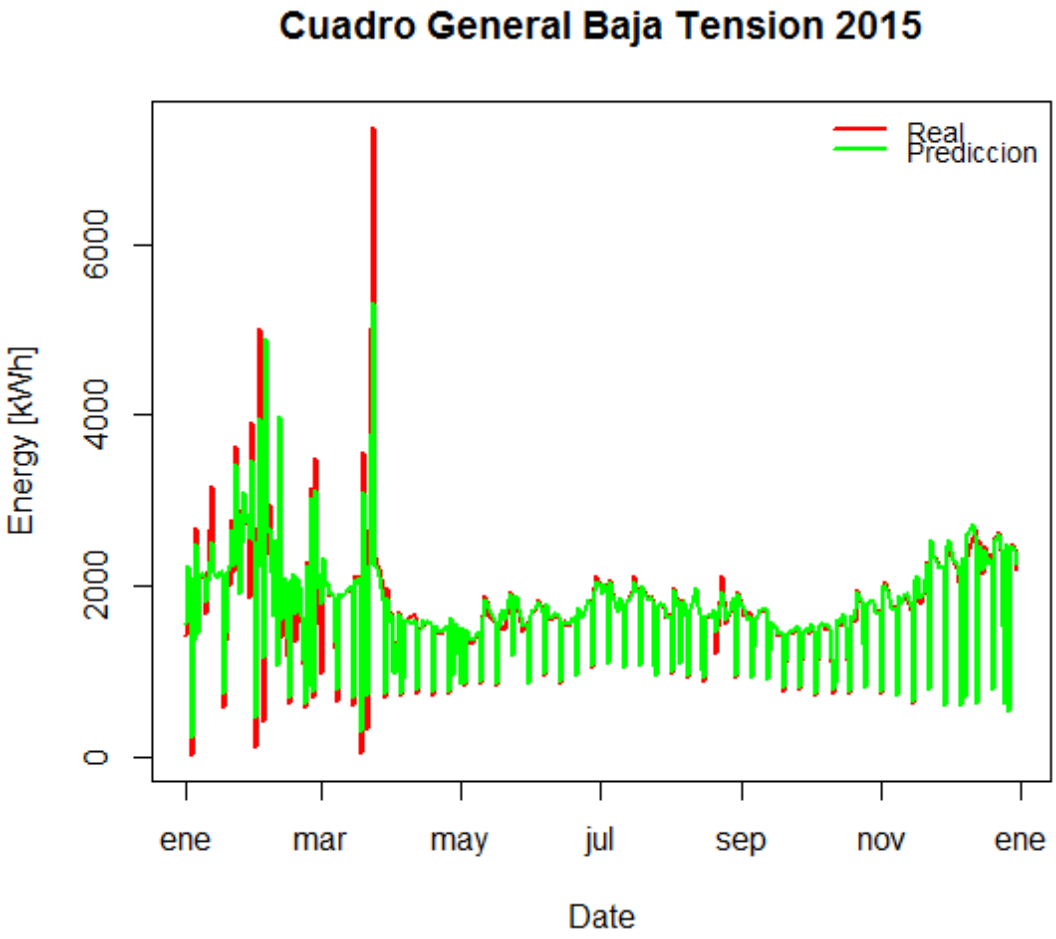
Gráfica 45: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales.

La siguiente técnica es utilizar un árbol de decisión.



Gráfica 46: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en fines de semana y restantes utilizando el conjunto de datos originales.

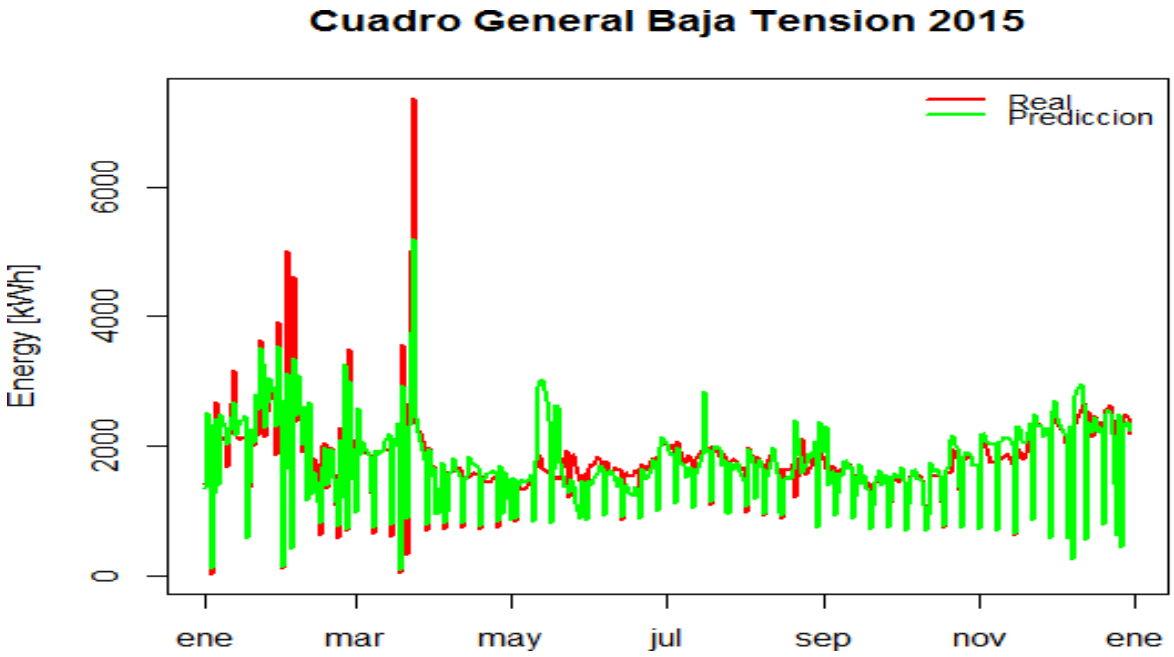
Por último probamos con un random forest.



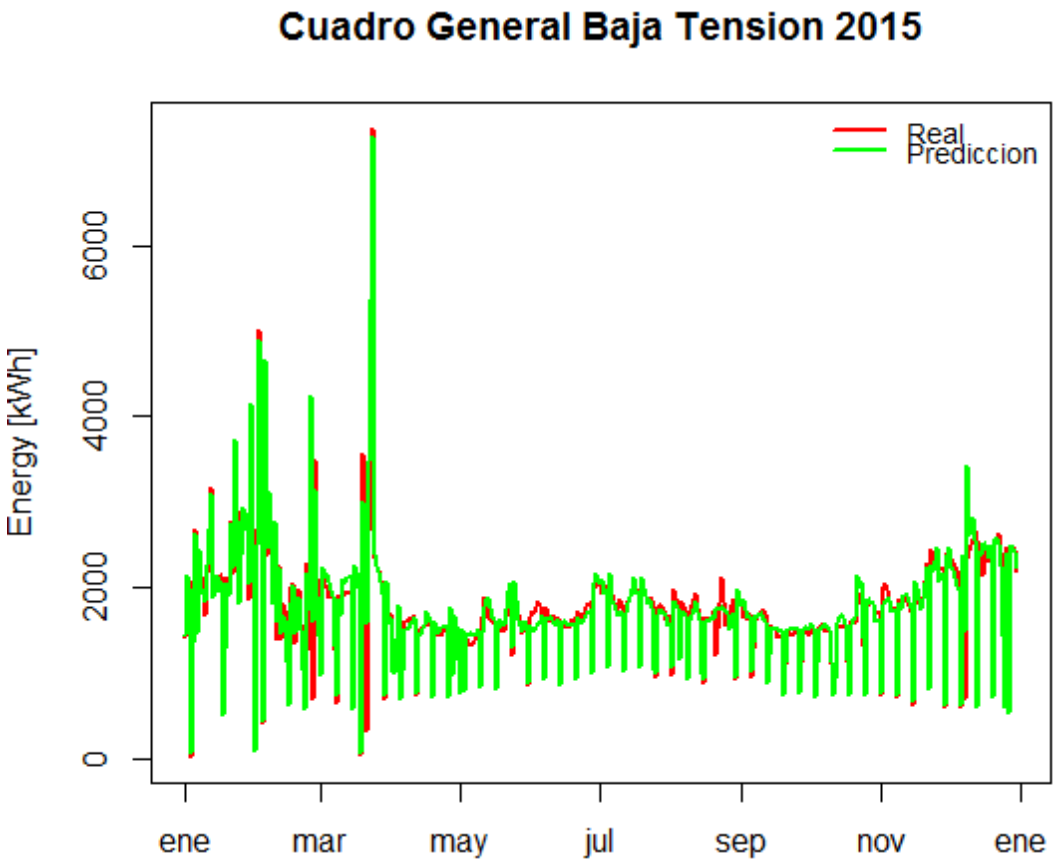
Gráfica 47: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en fines de semana y restantes y el conjunto de datos originales.

La segunda versión tiene en cuenta dos grupos también, pero esta vez uno que contenga los días laborales y otro los festivos.

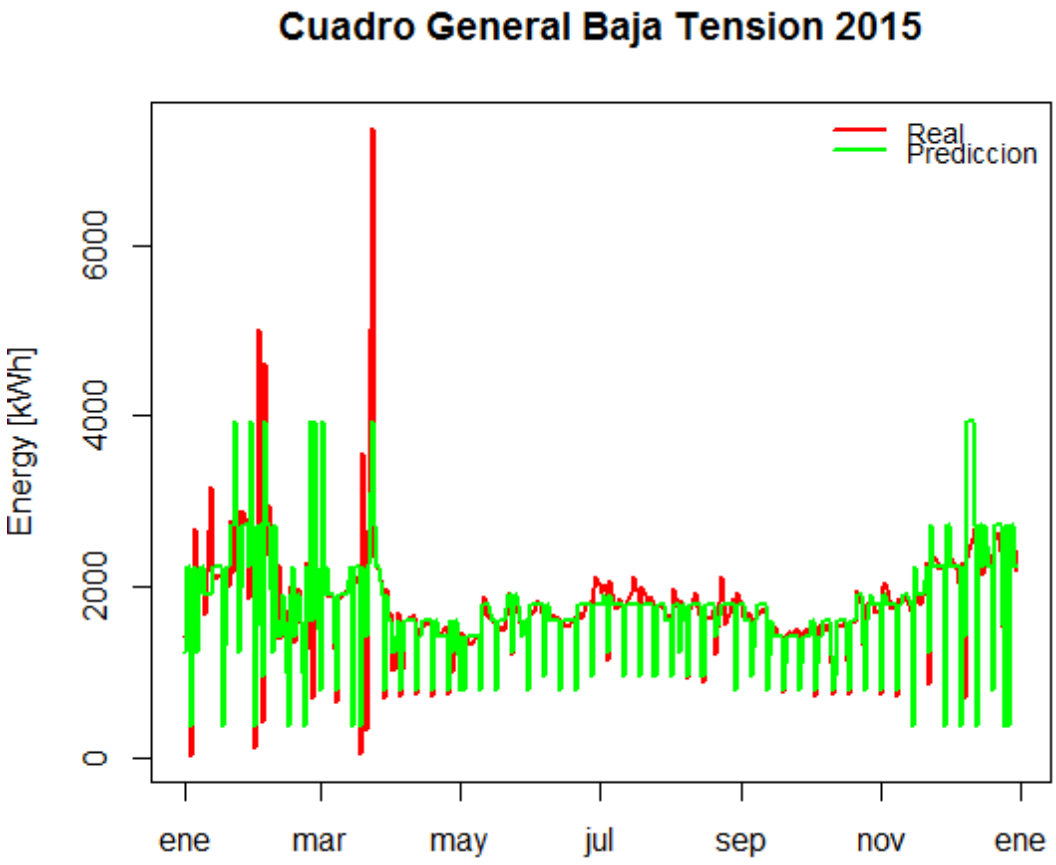
Así, aplicando los mismos algoritmos vistos anteriormente pero a los datos clasificados de otra forma, obtenemos las siguientes gráficas;



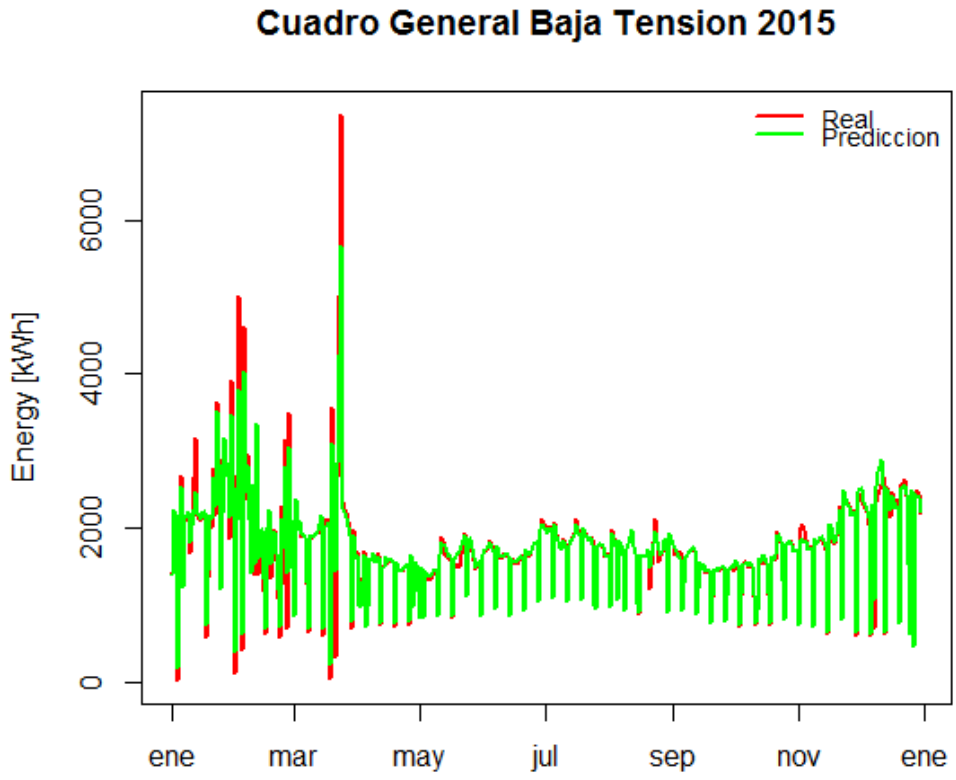
Gráfica 48: Resultados obtenidos mediante el uso de dos modelos de regresión lineal múltiple según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales.



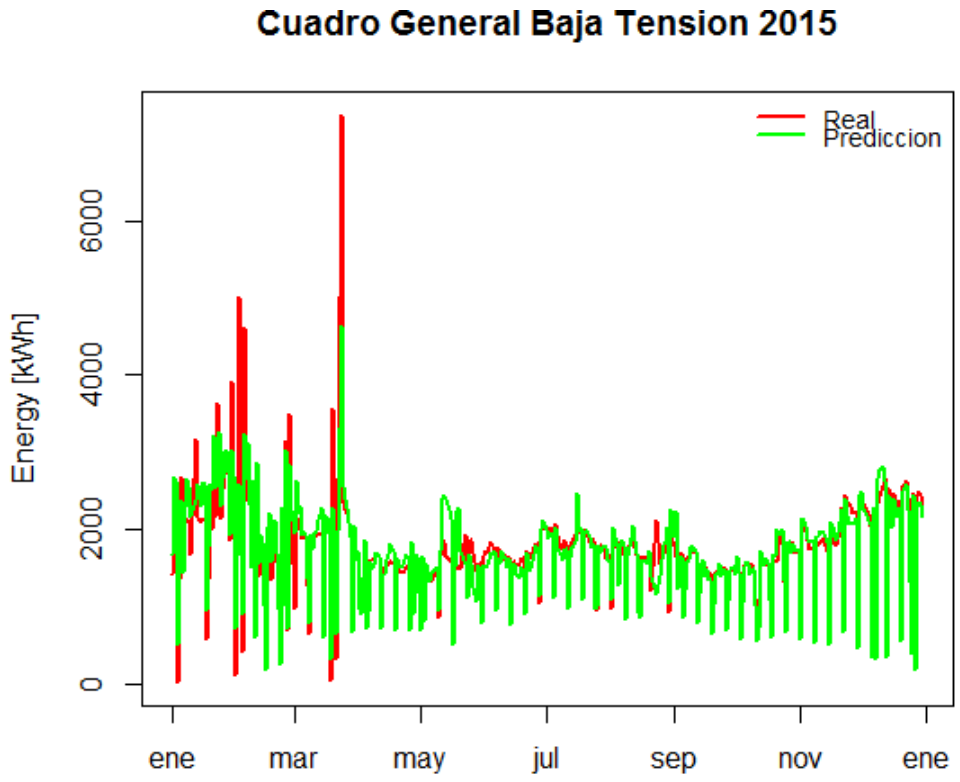
Gráfica 49: Resultados obtenidos mediante el uso de dos modelos de red neuronal según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales.



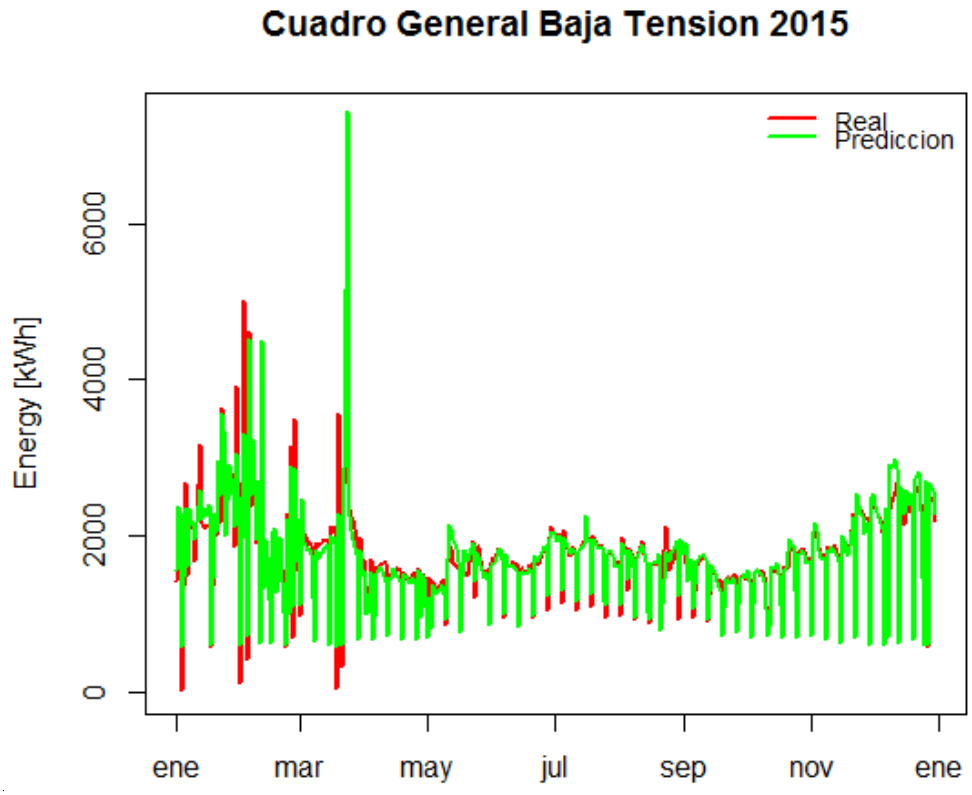
Gráfica 50: Resultados obtenidos mediante el uso de dos modelos de árbol de decisión según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales.



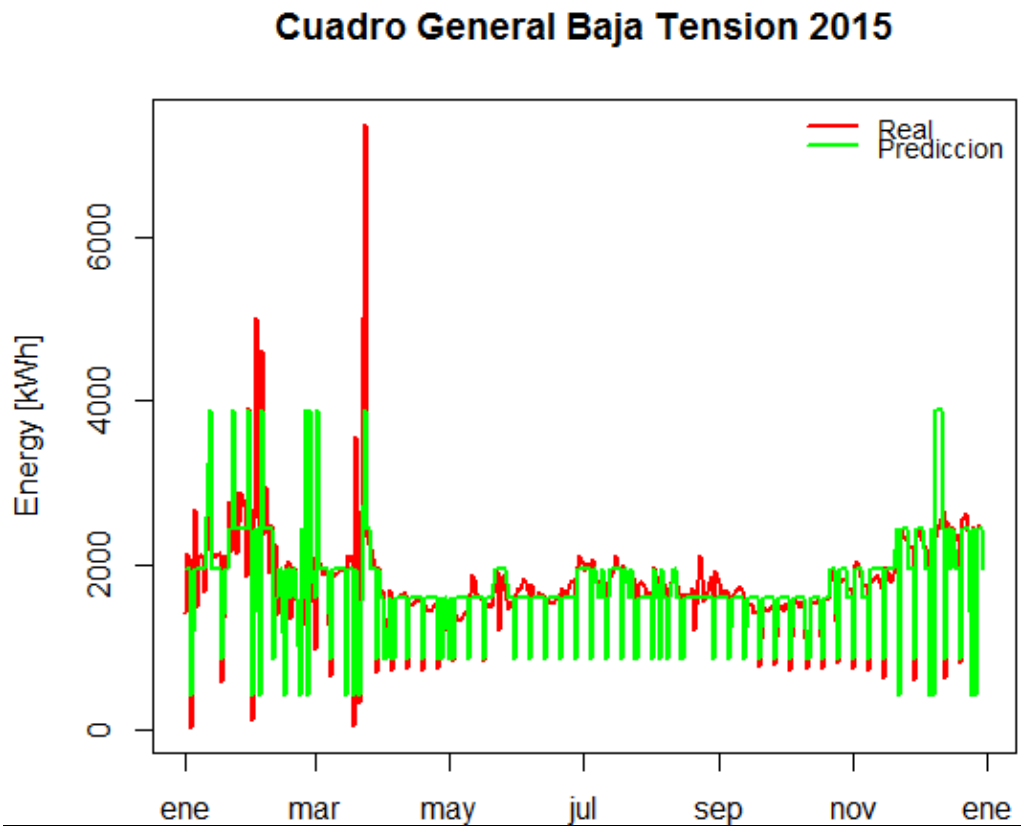
Gráfica 51: Resultados obtenidos mediante el uso de dos modelos de bosques aleatorios según la clasificación de datos en festivos y laborales utilizando el conjunto de datos originales.



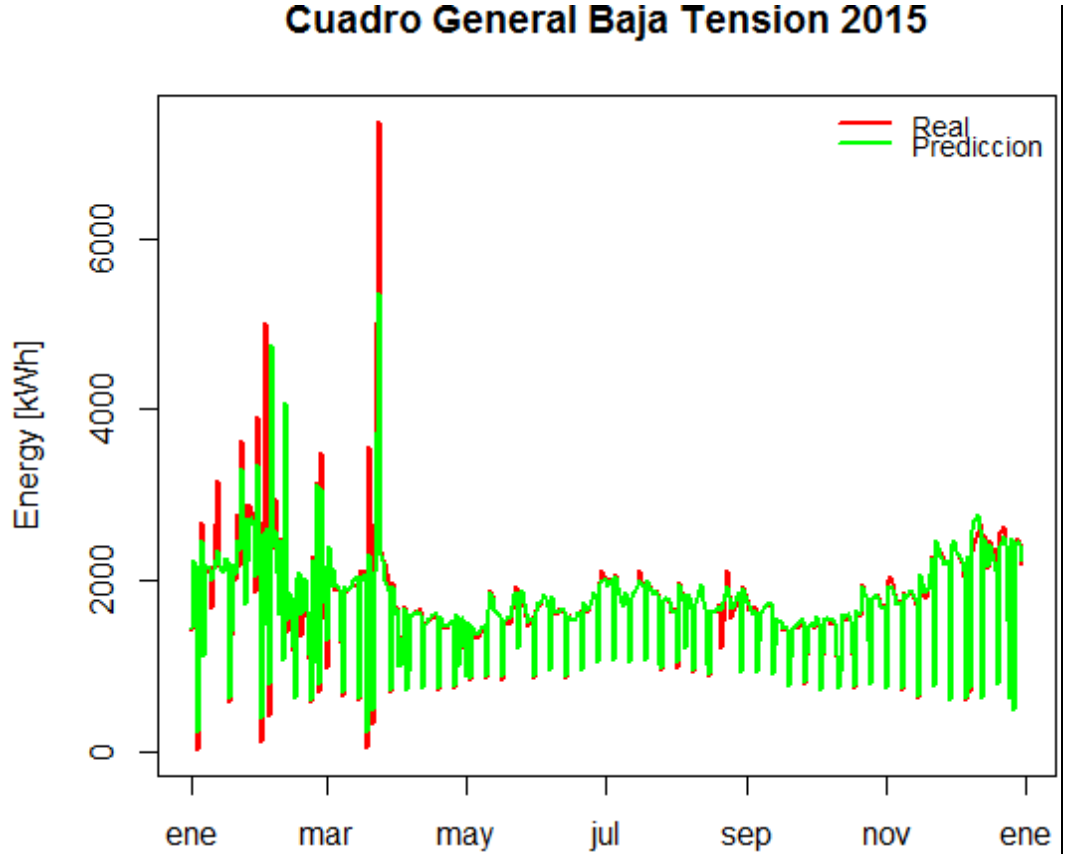
Gráfica 52: Resultados obtenidos mediante el uso de un modelo de regresión lineal múltiple utilizando el conjunto de datos originales y añadiendo el parámetro *laboralidad*.



Gráfica 53: Resultados obtenidos mediante el uso de un modelo de red neuronal utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad.



Gráfica 54: Resultados obtenidos mediante el uso de un modelo de árbol de decisión utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad.



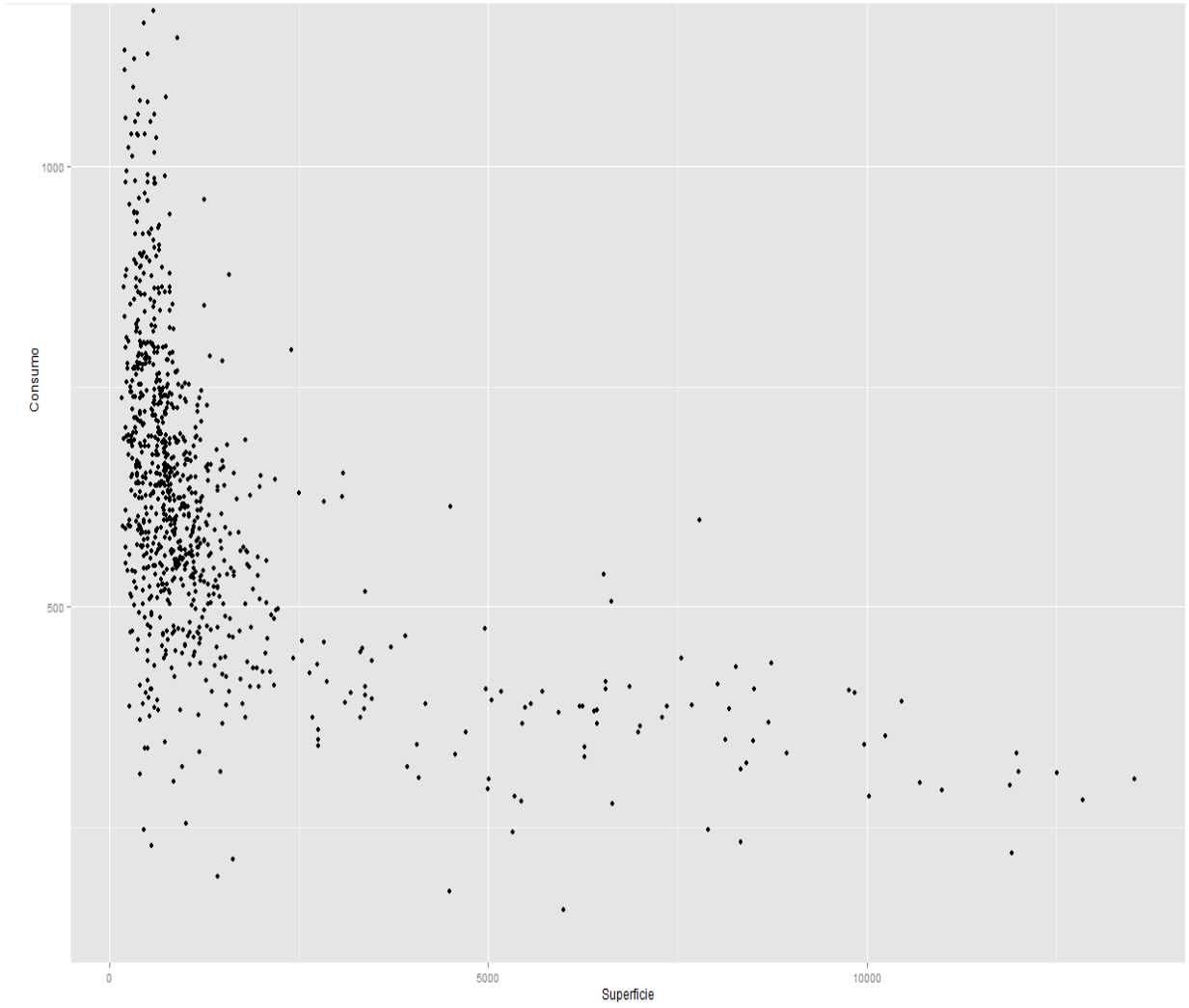
Gráfica 55: Resultados obtenidos mediante el uso de un modelo de bosques aleatorios utilizando el conjunto de datos originales y añadiendo el parámetro laboralidad.

8.2 ANEXO 2

A continuación se muestran varias capturas de la aplicación mencionada en la página 34.

El funcionamiento es simple; se muestra una gráfica en la que se representan distintos supermercados en España. En el eje X aparece la superficie del edificio en m^2 y en el eje Y el consumo medio diario en kWh.

Con el ratón podemos clicar en cualquier punto de la gráfica y en la parte inferior izquierda nos aparecerán en una tabla la dirección, superficie y consumo exactos de los supermercados más próximos al punto elegido. También se puede clicar y mantener para crear un cuadro que hará que aparezcan en la parte inferior derecha todos los supermercados y sus datos incluidos en dicho cuadro;



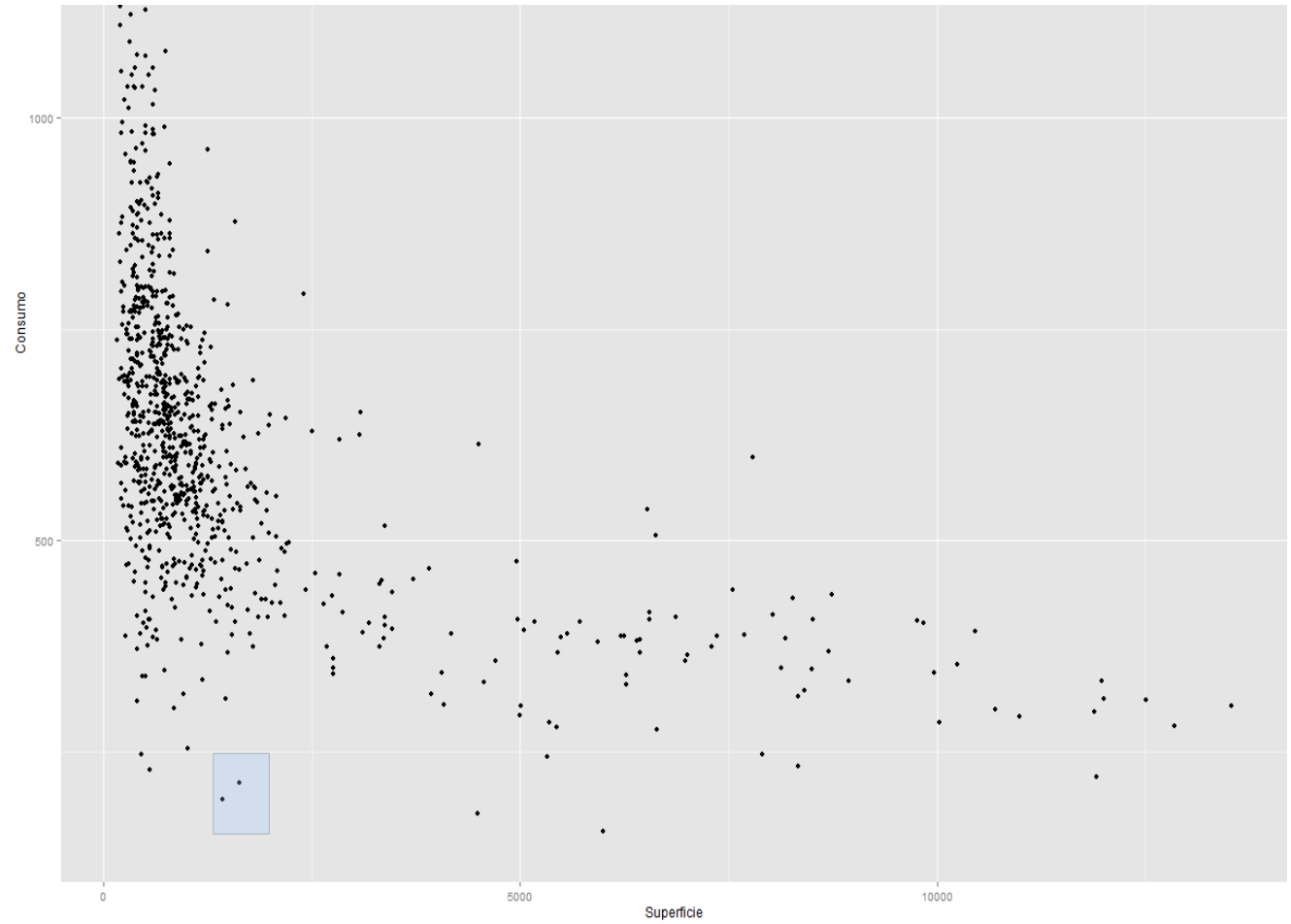
Supermercados cercanos

	Direccion	Superficie	Consumo	dist_
90	Araba, 41 - 20800 Zarauz (GIPUZKOA)	3377.00	517	0.76

Supermercados incluidos

Direccion	Superficie	Consumo

Figura 8: Ejemplo de uso de aplicación clicando en un punto.



Supermercados cercanos

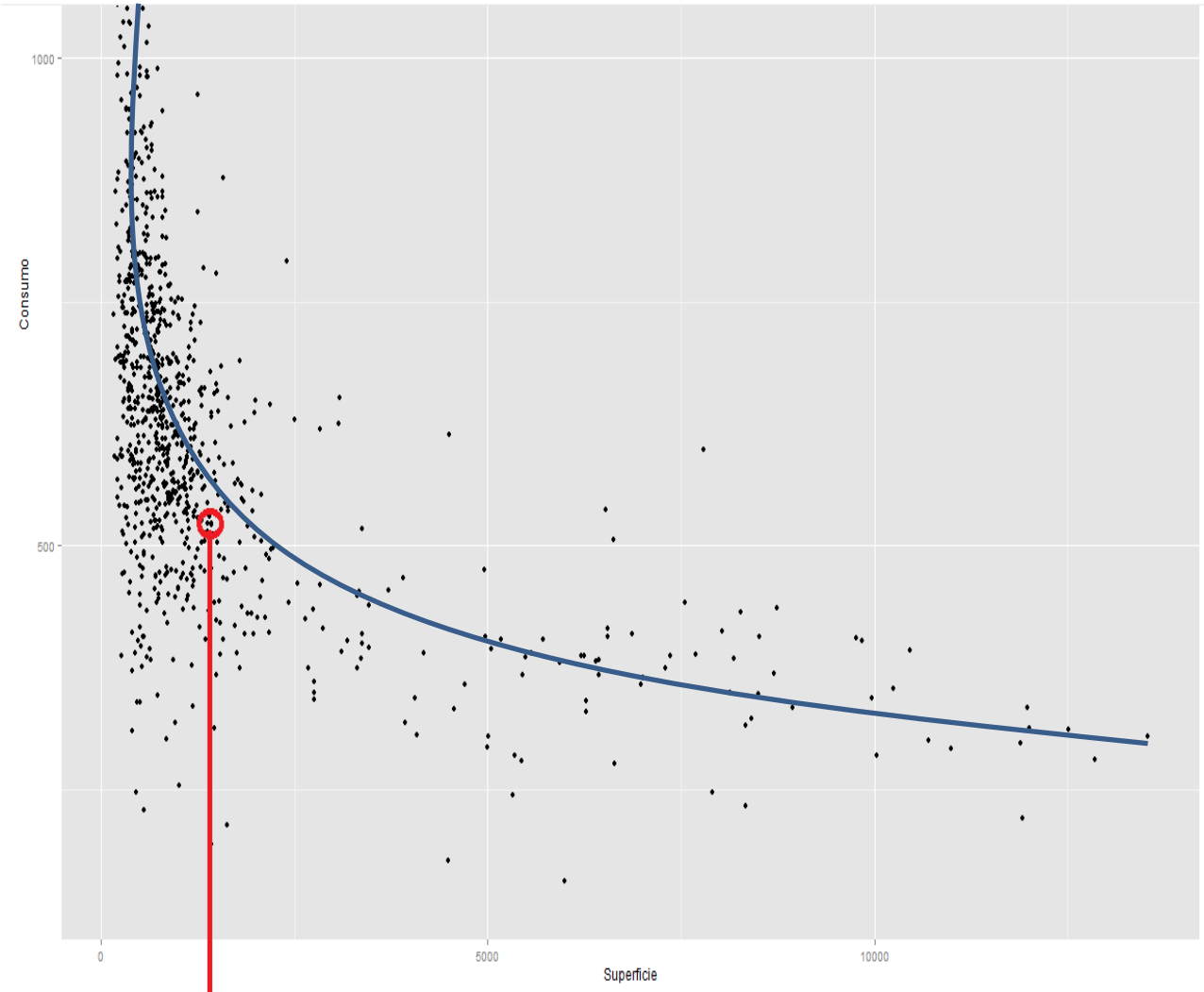
Direccion	Superficie	Consumo	dist_

Supermercados incluidos

	Direccion	Superficie	Consumo
63	SEGISMUNDO MORET 064, (ARTURO REYES)	1630.00	214
873	AV. SIERRA DE GREDOS 2	1423.00	194

Figura 9: Ejemplo de uso de la aplicación clicando y arrastrando.

Con esta herramienta se consigue observar y averiguar que supermercados son susceptibles de ser remodelados tal y como se hizo en el proyecto LIFE ZERO STORE de manera que se consiga un ahorro importante en el consumo energético. En la última figura podemos ver la zona correspondiente al supermercado que EROSKI eligió para ser sujeto de dicho proyecto.



Supermercados cercanos

	Direccion	Superficie	Consumo	dist_
510	C/ Jesus Arambarri s/n	1428.00	523	0.57
895	C/ BREMEN, 10, Bajo 80	1433.00	522	1.06
231	C/ RIO SOLIA 0 LOC	1431.00	522	1.09
834	FRANÇESC LAYRET 2 (MOLINS)	1382.00	523	4.20

Supermercados incluidos

Direccion	Superficie	Consumo

Figura 10: Dirección, superficie y consumo del Supermercado de EROSKI del proyecto LIFE ZERO STORE

8.3 ANEXO 3

Junto a este archivo podemos encontrar varios ficheros adjuntos, “Eneres_consumos.R”, “funciones.R” , “server.R” y “ui.R”, en los que está incluido el código utilizado con el software *R* para llevar a cabo cada una de las tareas y elementos mencionados en este proyecto.